The Shannon Switching Game

Stehouwer, Quinten

Quinten Stehouwer

The Shannon Switching Game

Master thesis

February 19, 2024

Thesis supervisor: prof. dr. F.M. Spieksma

Leiden University
Mathematical Institute

Contents

1 Introduction

In 1951, Claude E. Shannon, 'the father of information theory', built a machine that could play the two-player game on a graph nowadays known as the Shannon Switching Game, but which Shannon himself called Bird Cage.[1] A special case of the game has also been called (the game of) Gale (after David Gale, who independently invented the game) and Bridg-It, under which name the game was actually marketed [6, 8].

Shannon's machine was a physical resistance network in which the voltage was measured at different places to determine its moves. The players made their moves by turning switches on or off, effectively short-circuiting or cutting resistors. The name of the game, as well as the common names of the two players ('Short' and 'Cut'), originate here. The machine played well, but not perfectly [6].

In 1964, Alfred Lehman [12] used matroid theory to give necessary and sufficient conditions for the game to always be won (when played optimally) by respectively the Short player, the Cut player, or the player that goes first. Moreover, he formulated winning strategies for the player who could win. Both the characterization and the strategies, however, depend on the existence of two subsets of the edges that satisfy certain properties, and little was known about the construction or even the existence of such sets.

In 1969, John Bruno and Louis Weinberg [2] gave a constructive, graph-theoretic solution of the Shannon Switching Game, combining Lehman's results with work of Genya Kishi and Yoji Kajitani [10] about maximally distant trees and the principal partition of graphs.

This thesis gives an overview of the Shannon Switching Game, Lehman's [12] necessary and sufficient conditions for the game to be short, cut or neutral and the corresponding strategies, and an elaboration of Bruno and Weinberg's [2] application of Kishi and Kajitani's [10] results to finding a pair of disjoint cospanning trees in a graph if the game is short. The main contributions of this thesis are providing Lehman's results and their proofs with graphical examples to make them easier to read, and the elaboration of Bruno and Weinberg's algorithm mentioned above.

In Section 2 and Section 3 we respectively give a short explanation of the Shannon Switching Game, and classify instances of it. Since Lehman uses matroid theory to analyze the game, we first cover some basic matroid theory and Lehman's relevant lemmas in Section 4, and then translate the Shannon Switching Game on a graph to its matroidal version in Section 5. This then allows us to prove that the conditions that Lehman formulates are indeed necessary and sufficient for a game to be short, and we extract a winning strategy for Short from this proof in Section 6. After we have defined the dual matroid in Section 7, we analyze cut games in Section 8 in the same manner as we analyzed short games. For both short and cut games, we also formulate the winning strategy if the game is played on a graph. In Section 9 we explain how the described strategies can be applied to neutral games. Finally, in Section 10, we explain how one can find a pair of disjoint cospanning sets, which, together with Lehman's results, renders a constructive strategy for the Shannon Switching Game.

[1] It is not clear by whom the game itself was originally invented.

Games such as the Shannon Switching Game often have direct applications in network theory and, more generally, information theory [4]. More specifically, certain optimization problems could be reduced to (a variant of) such a game [13].

In future research, variants of the Shannon Switching Game could be studied. It would especially be interesting to see if the classifications of, and the strategies for, those variants are similar to those given by Lehman for the original game. As Lawler notes, '[i]t is not hard to devise variants of the switching game which are effectively unsolved' [11, p. 325]. He proposes a version where 'neither player is allowed to have more than k arcs[2] tagged at any time,' which is effectively the game of Bridg-It mentioned above, where both players have a limited number (20) of 'bridges'.

Some other ideas for variants (which fall beyond the scope of this thesis) are: one where players are allowed to 'untag' edges tagged by their opponent (and the opponent is not allowed to directly 'retag' that edge); one where the players may tag two (or more) edges per turn; one where there are multiple distinguished edges (which is to some degree discussed in [12]); one where the players may not tag two edges incident with the same vertex in a row; the directed variant discussed in [7].

2 Description of the Shannon Switching Game on a graph

The Shannon Switching Game is a two-player game that is played on a connected[3] graph G, one of whose edges $e_* = (u, v) \in E(G)$ is distinguished.[4] The two players - called either 'Short' and 'Cut' or 'the Short player' and 'the Cut player' - in turn tag edges $e \in E(G)$ with $e \neq e_*$, which then become either thick (tagged by the Short player) or dashed (tagged by the Cut player). An edge cannot be tagged more than once. The goals of the two players are different: the Short player aims to construct a path between u and v consisting only of thick edges. The Cut player aims to avoid this. The game finishes when one of the players reaches his or her goal (the Cut player winning when there is no possibility anymore of the Short player forming a path). Both players have complete information. By this description, the Shannon Switching Game can be considered a *combinatorial game*.[5] For a simple example of a game, see Figure 1.

[2] I.e. edges.

[3] The Shannon Switching Game can also be played on a disconnected graph. However, then only the connected component that contains e_* would be of interest. Because of this, we only consider connected graphs.

[4] In other descriptions of the game, this edge is sometimes only added when it is necessary to indicate the vertices u and v by means of an edge, which will be the case when we will translate the graphical situation into a matroidal one in Section 5. For simplicity, we require here that there always be such an edge.

[5] The definition of a combinatorial game varies: for example, Albert et al. [1, p. 3] demand that the winner is determined by who played last, which is not the case with the Shannon Switching Game.

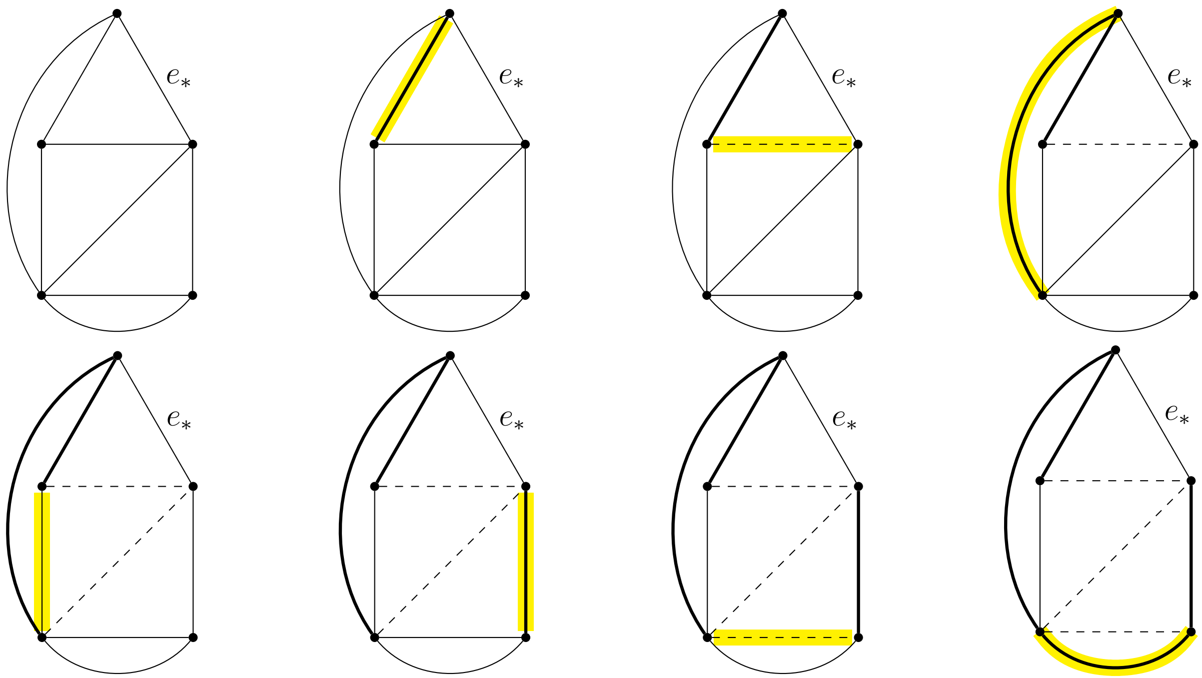


Figure 1: Example of a Shannon Switching Game. Edges tagged by Short are thick and edges tagged by Cut are dashed. The edge last played on is highlighted yellow.

3 Classification of Games: Short, Cut or Neutral

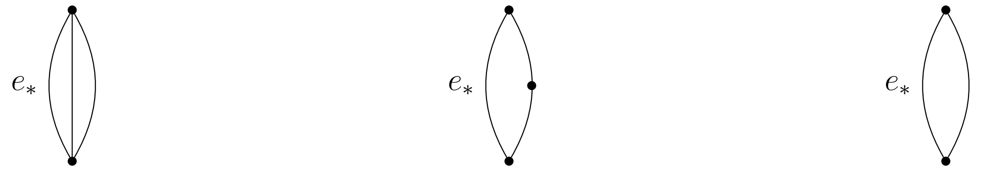
By the given description of the game, it is clear that one of the two players must win: when all edges are tagged, there either is a path between u and v consisting only of thick edges (a win for the Short player) or there is not (a win for the Cut player). We can now give the following classification of instances of the Shannon Switching Game (see Figure 2 for a simple example of every class):

(i) A game is *short* if the Short player, playing second, can win against any strategy of the Cut player.

(ii) A game is *cut* if the Cut player, playing second, can win against any strategy of the Short player.

(iii) A game is *neutral* if the player playing first can win against any strategy of the player playing second.

To see that these are indeed the only three options, we make the following observation. If a player, playing second, can win against any strategy of the player playing first, then it is also possible to win playing first by using the following strategy: as the first move, tag any edge. Then proceed by playing the same (winning) strategy as when playing second. If that strategy tells you to tag the edge you tagged first, tag any other (untagged) edge. This way, you play the same strategy as when playing second, only with an extra thick edge, which is never bad for you. It now logically follows that these three options classify all instances of the Shannon Switching Game.



(a) Short. (b) Cut. (c) Neutral.

Figure 2: Examples of a short, cut and neutral game.

4 Some matroid theory

Lehman [12] presents a winning strategy[6] for the Short player (if the game is short or if it is neutral and the Short player plays first[7]). He does this by first defining the Shannon Switching Game on a matroid instead of a graph. Then, he formulates three conditions that are necessary and sufficient for the game to be short and proves that they are. From the proof of the sufficiency, a winning strategy for the Short player becomes clear. Finally, he translates everything back to a graph-theoretical context.

To prove the necessity and sufficiency of the three conditions, Lehman first proves a series of lemmas. The proofs of some of these lemmas are quite concise. Hence, we here again give the lemmas and Lehman's proofs, but we present them in an extended (and hopefully clearer) way. Moreover, where both possible and helpful,[8] we illustrate (the proofs of) the lemmas with graphical examples. We would like to emphasize that these examples do not aim to fully depict (the proofs of) the lemmas, not merely because they are examples, but because a matroid is a much more general object than a graph, and not every matroid is a graphical one. However, precisely because matroids are so general and hence abstract, and because graphs are relatively easy to interpret, we believe that here the examples offer a valuable contribution. Moreover, the Shannon Switching Game is originally about graphs, so they seem to be a logical choice of representation.

We will start by giving some basic definitions concerning matroids, and then formulate, prove and illustrate Lehman's lemmas. In Section 5, we present his proof of the sufficiency and necessity of the three conditions for a game to be short.

4.1 Basic definitions

A matroid can be defined in many (equivalent) ways, one of them being in terms of circuits. Since the Shannon Switching Game is essentially about circuits/cycles[9], this is the one that we will be working with (and the one that Lehman uses).

[6] As far as we know, no general *optimal* strategy is known, in the sense that it wins the game in as few turns as possible.

[7] See Section 9 for a more detailed explanation of how the strategy is applied to neutral games.

[8] In particular, we chose not to illustrate (the proof of) Lemma 6. Since the proof consists of many steps, involves edge-dependent subsets for every edge in the graph, and contains complex subsets (in that they consist of many subsets themselves), we believe that any illustration of (the proof of) the lemma would either be very lengthy, or would not illustrate all important facets of it.

[9] The path that Short aims to form, together with e_*, constitutes a cycle in the graph.

A *matroid* M is a pair $(E, \mathfrak{C})$, with E a finite non-empty set and $\mathfrak{C} \subseteq 2^E$ a collection of subsets of E respecting the following properties:

1. $\emptyset \notin \mathfrak{C}$.
2. If $C', C'' \in \mathfrak{C}$ and $C' \subseteq C''$, then we have $C' = C''$.
3. If $C', C'' \in \mathfrak{C}$, $e \in C' \cap C''$ and $C' \neq C''$, then there exists $C \in \mathfrak{C}$ such that $C \subseteq C' \cup C'' - \{e\}$.

Here, we denote set difference by $-$. The set operations used in this text ($\cup$, $\cap$ and $-$) are to be performed from left to right, unless otherwise indicated by brackets.

An element $C \in \mathfrak{C}$ is called a *circuit* and elements $e \in E$ are called *branches*.

The *span* $\mathrm{sp}(A)$ of a subset $A \subseteq E$ is defined as

$$\mathrm{sp}(A) := \{e \in E \mid e \in A \text{ or there exists a circuit } C \in \mathfrak{C} \text{ such that } e \in C \subseteq A \cup \{e\}\}$$

and A is said to span a branch $e \subseteq E$ if $e \in \mathrm{sp}(A)$.

To illustrate these concepts, we look at a few matroids. The graphical matroid $M(G) = (E, \mathfrak{C})$ of a graph G has the edges of G as its branches and the cycles[10] in G as its circuits. To see that $M(G)$ is indeed a matroid, note that the empty set is not a cycle, cycles do not contain other cycles and if we have two different cycles with a non-empty intersection, the symmetrical difference of those two cycles contains (actually, is) again a cycle. For example, if G is the graph in Figure 3, we have $\mathfrak{C} = \{\{1,3,4\}, \{1,2,4,5\}, \{1,2,4,6,7\}, \{2,3,5\}, \{2,3,6,7\}, \{5,6,7\}\}$. We also have, for example, $\mathrm{sp}(\{2,3,7\}) = \{2,3,5,6,7\}$.

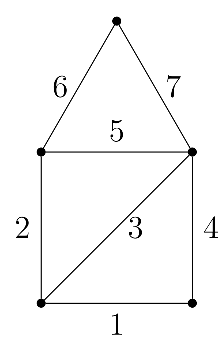
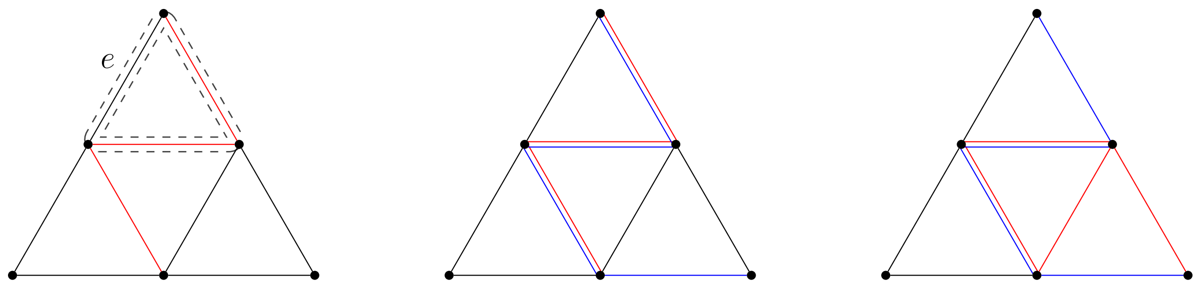


Figure 3: A graph.

A matroid is, however, not necessarily based upon a graph. For example, consider the set $A = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\} \subseteq \mathbb{R}^5$ with $a_1 = (1,1,0,0,0)^\top, a_2 = (1,0,1,0,0)^\top, a_3 = (1,0,0,1,0)^\top, a_4 = (0,1,0,1,0)^\top, a_5 = (0,0,1,1,0)^\top, a_6 = (0,0,1,0,1)^\top, a_7 = (0,0,0,1,1)^\top$. The vector matroid $M(A) = (E, \mathfrak{C})$ of the set of vectors A has the vectors in A as its branches and the minimal linearly dependent subsets of A as its circuits. This definition again satisfies the three properties of a matroid, where the last property follows from basic linear algebra (the 'exchange property'). The matroids $M(A)$ and $M(G)$ are isomorphic[11]: every vector

[10] That is, the circuits in which only the first and the last vertices are equal. They are also called simple circuits.

[11] See, for example, [9, p. 235].



(a) Statements (ii) and (iv). (b) Statement (iii). (c) Statement (v).

Figure 4: Examples of the statements in Figure 4. Edges in A are red, edges in B are blue, and C is gray and dashed.

a_i corresponds with an edge i, and the vector entries correspond with the vertices in such a way that the entries equal to 1 correspond with the end points of the corresponding edge. A subset of the vectors is now precisely minimal dependent if the corresponding edges form a cycle: their linear combination is then equal to zero, since every vertex in the cycle is the endpoint of exactly 2 vertices. Hence, $M(A)$ is also a graphical matroid.

An example of a matroid that is not graphical, is $M = (E, \mathfrak{C})$ with $E = \{1, 2, 3, 4\}$ and $\mathfrak{C} = \{\{2, 3, 4\}, \{1, 3, 4\}, \{1, 2, 4\}, \{1, 2, 3\}\}$ - the so-called 2-uniform matroid: a graph that has cycles $\{2, 3, 4\}$ and $\{1, 3, 4\}$ must also contain the cycle $\{1, 2\}$, but $\{1, 2\} \notin \mathfrak{C}$.

4.2 Some lemmas about matroids

The following six lemmas are from Lehman [12]. Lemma 1 corresponds to Lemmas 3, 4, 5, 6 and 7 from [12]; Lemmas 2, 3, 4, 5 and 6 correspond to Lemmas 8, 9, 11, 12 and 13, respectively. We have included only the most basic lemmas (summed up in Lemma 1) and the lemmas that are used in the proof of Lehman's Theorem 14 (our Theorem 1).

Lemma 1. *Let $M = (E, \mathfrak{C})$ be a matroid, $A, B \subseteq E$ subsets of the set of branches and $e \in E$ a branch. Then the following statements hold:*

(i) $A \subseteq \mathrm{sp}(A)$.

(ii) $e \in \mathrm{sp}(A) - A$ *if and only if there exists a circuit C such that $C - A = \{e\}$.*

(iii) If $A \subseteq B$ *then* $\mathrm{sp}(A) \subseteq \mathrm{sp}(B)$.

(iv) $\mathrm{sp}(A) = \mathrm{sp}(\mathrm{sp}(A))$.

(v) If $A \subseteq \mathrm{sp}(B)$ *then* $\mathrm{sp}(A) \subseteq \mathrm{sp}(B)$.

We omit the proof, since the results are quite basic.[12] To get a sense for why they are true, see Figure 4.

Lemma 2. *Let $a \in E$ be a branch and $A, B \subseteq E$ subsets of branches, such that $a \in A - B$ and* $\mathrm{sp}(B) \subseteq \mathrm{sp}(A)$. *Then either* $\mathrm{sp}(B) \subseteq \mathrm{sp}(A - \{a\})$ *or there exists a branch $b \in B - A$*

[12] Moreover, we will not refer to this lemma every time we use it, since this would decrease readability.

such that $\text{sp}(B) \subseteq \text{sp}(A \cup \{b\} - \{a\})$.

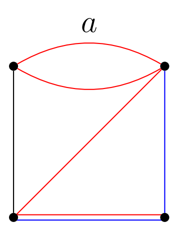
Proof. We distinguish between two cases.

1. See Figure 5a. If $B - A \subseteq \text{sp}(A - \{a\})$, then $B \subseteq \text{sp}(A - \{a\})$ and hence by Lemma 1(v) we have $\text{sp}(B) \subseteq \text{sp}(A - \{a\})$.

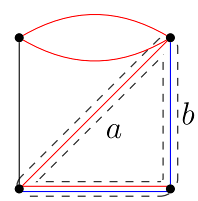
2. See Figure 5b. If $B - A \not\subseteq \text{sp}(A - \{a\})$, then there exists a branch $b \in (B - A) - \text{sp}(A - \{a\})$. Since $b \in B \subseteq \text{sp}(A)$, there exists a circuit $C \subseteq A \cup \{b\}$ with $b \in C$. Since $b \notin \text{sp}(A - \{a\})$, there is no circuit in $A - \{a\} \cup \{b\}$ that contains b and hence $a \in C$. Then $a \in \text{sp}(A \cup \{b\} - \{a\})$, so $A \subseteq \text{sp}(A \cup \{b\} - \{a\})$ and hence $\text{sp}(B) \subseteq \text{sp}(A) \subseteq \text{sp}(A \cup \{b\} - \{a\})$.

□

Intuitively, in the second case, b 'repairs' A, in the sense that $A \cup \{b\} - \{a\}$ spans B again.



(a) The case $B - A \subseteq \text{sp}(A - \{a\})$.



(b) The case $B - A \not\subseteq \text{sp}(A - \{a\})$.

Figure 5: Illustration of Lemma 2. Edges in A are red, edges in B are blue, and C is gray and dashed.

Lemma 3. *Let $A, B \subseteq E$ be subsets of the set of branches and $a, d \in E$ branches, such that $a \in A - B$, $d \notin \text{sp}(A \cap B)$ and $d \in \text{sp}(A) = \text{sp}(B)$. Then there exists a branch $b \in B - \text{sp}(A \cap B)$ such that $\text{sp}(A \cup \{b\} - \{a\}) = \text{sp}(B)$.*

Proof. Since $d \in \text{sp}(B) - \text{sp}(A \cap B)$, $B - \text{sp}(A \cap B)$ is not empty: suppose it was, then B would be spanned by $A \cap B$ and hence $d \in \text{sp}(B) \subseteq \text{sp}(A \cap B)$. We distinguish between two cases:

1. See Figure 6a. If $\text{sp}(A - \{a\}) = \text{sp}(B)$, any $b \in B - \text{sp}(A \cap B)$ suffices.

2. See Figure 6b. If $\text{sp}(A - \{a\}) \neq \text{sp}(B)$, we have $\text{sp}(B) \not\subseteq \text{sp}(A - \{a\})$, so from Lemma 2 it follows that there exists a branch $b \in B - A$ such that $\text{sp}(B) \subseteq \text{sp}(A \cup \{b\} - \{a\})$ and even $\text{sp}(B) = \text{sp}(A \cup \{b\} - \{a\})$, since $\text{sp}(A - \{a\}) \subseteq \text{sp}(A) = \text{sp}(B)$. Moreover, $b \notin \text{sp}(A - \{a\})$ (otherwise we would have $\text{sp}(A - \{a\}) = \text{sp}(A \cup \{b\} - \{a\}) = \text{sp}(B)$, which is against the assumption), so $b \in B - \text{sp}(A - \{a\}) \subseteq B - \text{sp}(A \cap B)$.

□

This result is quite similar to Lemma 2, the most important difference being that there now *always* exists a branch $b \in B$ that repairs A (if there is anything to repair), and we can

(a) The case $\mathrm{sp}(A - \{a\}) = \mathrm{sp}(B)$. (b) The case $\mathrm{sp}(A - \{a\}) \neq \mathrm{sp}(B)$.

Figure 6: Illustration of Lemma 3. Edges in A are red and edges in B are blue.

demand that $b \notin \mathrm{sp}(A \cap B)$. In the proof of Theorem 1, this will guarantee that the Short player will always have a branch b to play on in response to a play on a by the Cut player.

Lemma 4. *Let $A, B \subseteq E$ be subsets of the set of branches and $a, d \in E$ branches such that $a \in A \cap B$, $d \notin A \cup B$ and $d \in \mathrm{sp}(A) = \mathrm{sp}(B)$. Then there exist $A', B' \subseteq E$ such that*

(i) $A' \cup B' \subseteq A \cup B$;

(ii) $A' \cap B' \subseteq A \cap B - \{a\}$;

(iii) $\mathrm{sp}(A' \cup \{d\}) = \mathrm{sp}(B')$.

Proof. For every $e \in (B - A) \cup \{d\}$, let C_e^A be a circuit with $C_e^A \subseteq A \cup \{e\}$ and $e \in C_e^A$ (which exists because $e \in \mathrm{sp}(A) - A$). Similarly, for every $e \in (A - B) \cup \{d\}$, let C_e^B be a circuit with $C_e^B \subseteq B \cup \{e\}$ and $e \in C_e^B$. See Figure 7.

Define[13] the sequence $B_0, A_1, B_1, A_2, B_2, ...$ by $B_0 = \emptyset$, $A_1 = C_d^A - \{d\}$,

$$A_{i+1} = A_i \cup ((B_i - \mathrm{sp}(A_i)) \cap A) \cup (\bigcup_{e \in (B_i - \mathrm{sp}(A_i)) - A} (C_e^A - \{e\})),$$

$$B_{i+1} = B_i \cup ((A_{i+1} - \mathrm{sp}(B_i)) \cap B) \cup (\bigcup_{e \in (A_{i+1} - \mathrm{sp}(B_i)) - B} (C_e^B - \{e\})).$$

In other words, to construct, say, A_{i+1}, we take A_i and for every $e \in B_i$ that was not already spanned by A_i, we either add e itself (if $e \in A$), or we add $C_e^A - \{e\}$ (if $e \notin A$). Since E is finite and we have $A_i \subseteq A_{i+1}$ and $B_i \subseteq B_{i+1}$ for every i, $A_\infty := \lim_{i \to \infty} A_i$ and $B_\infty := \lim_{i \to \infty} B_i$ exist. See Figure 8.

We now distinguish between two cases.

1. If $a \notin A_\infty \cap B_\infty$, set $A' = A_\infty$ and $B' = B_\infty$. From the construction of the sequence it is clear that we then have $d \in \mathrm{sp}(A') = \mathrm{sp}(B')$ and $A' \subseteq A$ and $B' \subseteq B$. Hence, $A' \cup B' \subseteq A \cup B$, $A' \cap B' \subseteq A \cap B - \{a\}$ and $\mathrm{sp}(A' \cup \{d\}) = \mathrm{sp}(B')$.

[13] In [12] there is a typo in the definition of B_{i+1}. (The second $\cap$ should be a $\cup$.)

2. If $a \in A_\infty \cap B_\infty$, let k be the minimal index such that $a \in B_k - A_k$ or $a \in A_{k+1} - B_k$.[14] We then have one of the following three situations:

   (i) If $a \in A_1$, set $A' = A - \{a\}$ and $B' = B$.

   (ii) If $a \in B_k - A_k$, then there exists a branch $b \in (A_k - \text{sp}(B_{k-1})) - B$ such that $a \in C_b^B - \{b\}$.[15] To see this, note that B_k is the union of three sets. a cannot be in one of the first two, i.e. B_{k-1} or $(A_k - \text{sp}(B_{k-1})) \cap B$: if $a \in B_{k-1}$, then k would not have been minimal, and if $a \in (A_k - \text{sp}(B_{k-1})) \cap B$, we would not have $a \in B_k - A_k$. Now b, A and $B \cup \{b\} - \{a\}$ satisfy the original assumptions for a, A and B: $b \in A \cap (B \cup \{b\} - \{a\})$, $d \notin A \cup (B \cup \{b\} - \{a\})$ and $d \in \text{sp}(A) = \text{sp}(B \cup \{b\} - \{a\})$. Hence we can go through the same process, using the same C_e^A's and C_e^B's, except setting C_a^B to be the previous C_b^B - we do not define them again, so even though new circuits might have been created, we do not accept those. We do, however, redefine the sequence $B_0, A_1, B_1, A_2, B_2, \ldots$. Note that we still have $b \in A_k - B_{k-1}$. We now have situation (i) or (iii) - where b takes the role of a, A the role of A, and $B \cup \{b\} - \{a\}$ the role of B - and can proceed accordingly.

   (iii) If $a \in A_{k+1} - B_k$, the situation is analogous to the above: there exists a branch $b \in (B_k - \text{sp}(A_k)) - A$ such that $a \in C_d^A - \{b\}$. Now b, $A \cup \{b\} - \{a\}$ and B satisfy the original assumptions for a, A and B. Hence we can go through the same process, using the same C_e^A's and C_e^B's, except setting C_a^A to be the previous C_b^A. We now have situation (ii) - where b takes the role of a, $A \cup \{b\} - \{a\}$ the role of A, and B the role of B - and can proceed accordingly.

   At every step, the branch b that replaces a is in a set further to the left in the sequence; hence it is clear that the above process is finite and that eventually we will arrive at situation (i). A' and B' then clearly meet the required conditions. See Figure 9.

□

In the proof of Theorem 1, this lemma allows for the replacement of b in A_a by e_* in $A'_a \cup \{e_*\}$.[16]

Since the proof above is quite nontransparent, we will explain the idea behind it. After defining the circuits and the sequence, if $a \in B_k - A_k$ or $a \in A_{k+1} - B_k$ for some $k \geq 1$, we perform an iterative process of 'reshuffling' the sets A and B to achieve that a is not in A' anymore, and d 'takes its role': we find a branch b so that a is in C_b^A (or in C_b^B), we add b to B and we remove a from B. We then perform the same steps, b taking the role of a, and we keep doing this procedure until the replacing branch is in A_1.

Lemma 5. *Let $A, B \subseteq E$ be subsets of the set of branches and $a, d \in E$ branches such that $a \in A - B$, $d \notin A \cup B$ and $d \in \text{sp}(A) = \text{sp}(B)$. Then there exist $A', B' \subseteq E$ such that*

[14] Lehman [12, p. 693] seems to claim that k (or his $|a|$) is unique(ly defined). The example in the appendix shows that this is not necessarily true.

[15] Since we found this step a little counterintuitive ourselves, we would like to point out to the reader that we are here picking a branch b such that a is in $C_b^B - \{b\}$, and not a branch b such that b is in some circuit dependent on a.

[16] For technichal reasons, e_* is not in A'_a and instead we work with $A'_a \cup \{e_*\}$.

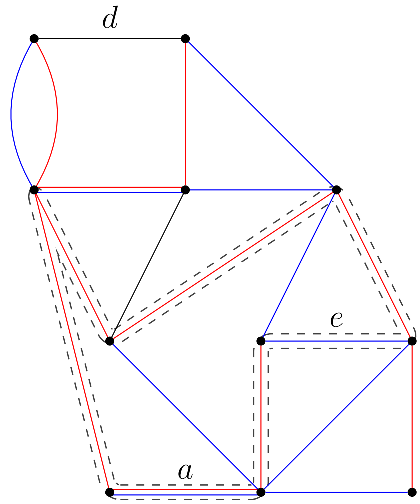


Figure 7: An example of the situation in Lemma 4, including a circuit C_e^A. Edges in A are red, edges in B are blue, and C_b^A is gray and dashed.

(i) $A' \cup B' \subseteq A \cup B - \{a\}$;

(ii) $A' \cap B' \subseteq A \cap B$;

(iii) $\mathrm{sp}(A' \cup \{d\}) = \mathrm{sp}(B')$.

Proof. We distinguish between two cases.

1. If $\mathrm{sp}(A - \{a\}) = \mathrm{sp}(B)$, set $A' = A - \{a\}$ and $B' = B$.
2. If $\mathrm{sp}(A - \{a\}) \neq \mathrm{sp}(B)$, we have $\mathrm{sp}(B) \nsubseteq \mathrm{sp}(A - \{a\})$ and hence by Lemma 2 there exists a branch $b \in B - A$ such that $\mathrm{sp}(B) \subseteq \mathrm{sp}(A \cup \{b\} - \{a\})$ and thus $\mathrm{sp}(B) = \mathrm{sp}(A \cup \{b\} - \{a\})$. We now have $b \in (A \cup \{b\} - \{a\}) \cap B$ and clearly $d \notin (A \cup \{b\} - \{a\}) \cup B$ and $d \in \mathrm{sp}(A \cup \{b\} - \{a\}) = \mathrm{sp}(B)$, so by Lemma 4 there exist $A', B' \subseteq E$ such that
   - $A' \cup B' \subseteq (A \cup \{b\} - \{a\}) \cup B \subseteq A \cup B$,
   - $A' \cap B' \subseteq (A \cup \{b\} - \{a\}) \cap B - \{b\} \subseteq A \cap B$ and
   - $\mathrm{sp}(A' \cup \{d\}) = \mathrm{sp}(B')$.

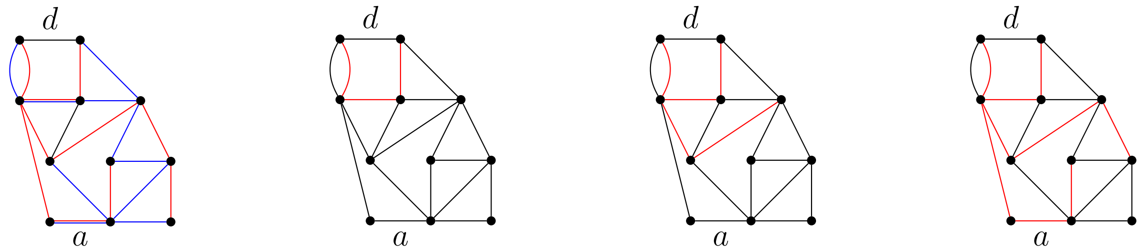
   Since $a \notin A', B'$, we even have $A' \cup B' \subseteq A \cup B - \{a\}$

□

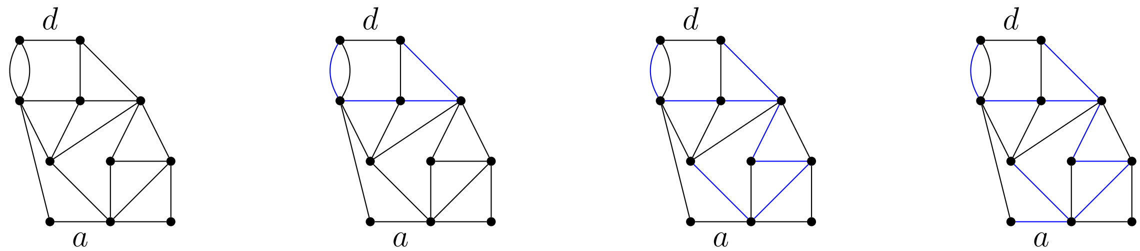
Lemma 5 is almost identical to Lemma 4, the only difference being that $a \in A - B$ instead of $a \in A \cap B$.

Lemma 6. *Let $d \in E$ be a branch such that $E - \{d\} \neq \emptyset$ and suppose that for every branch $a \in E - \{d\}$ there exist $A_a, B_a \subseteq E$ such that $a, d \notin A_a \cup B_a$ and $\mathrm{sp}(A_a \cup \{d\}) = \mathrm{sp}(B_a)$. Then there exist $A, B \subseteq E$ such that*

(i) $d \notin A \cup B$;



(a) A copy of the situation in Figure 7. (b) A_1 (c) A_2 (d) $A_3 = A_\infty$



(e) B_0 (f) B_1 (g) B_2 (h) $B_3 = B_\infty$

Figure 8: The construction of the sequence. For every i, edges in A_i are red and edges in B_i are blue.

(ii) $A \cap B \subseteq \bigcup_{a \in E-\{d\}} (A_a \cap B_a)$*;*

(iii) $d \in \mathrm{sp}(A) = \mathrm{sp}(B)$.

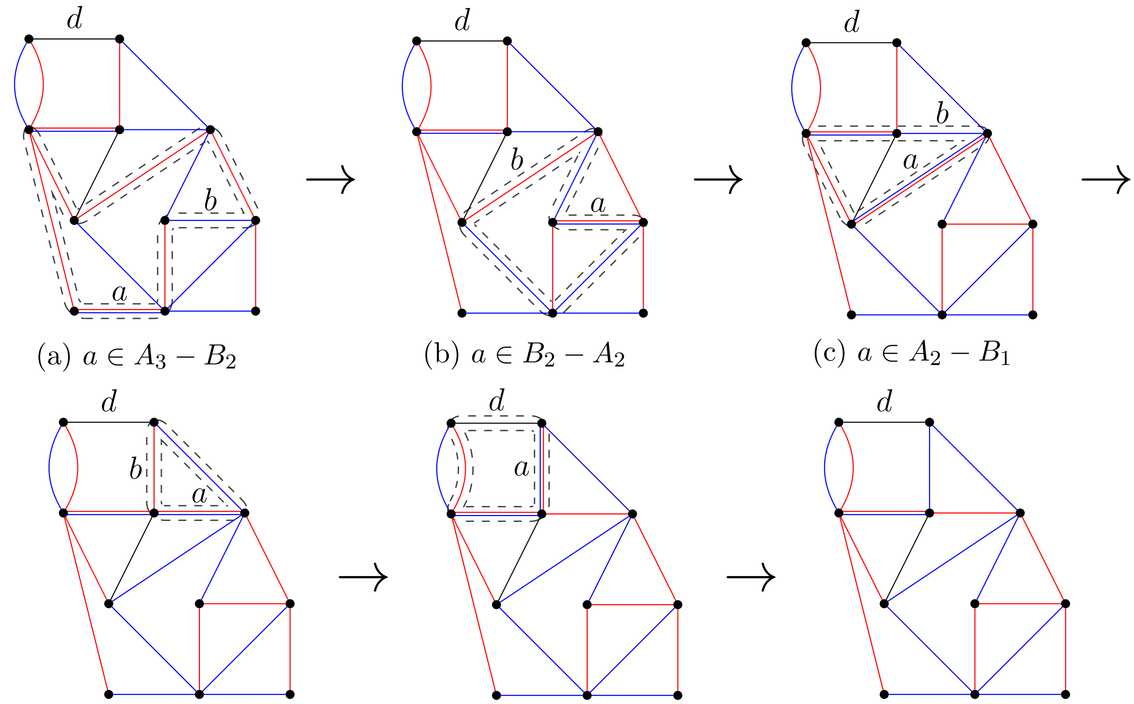
Proof. If, for some $a \in E - \{d\}$, we have $d \in \mathrm{sp}(A_a)$, set $A = A_a$ and $B = B_a$. Otherwise, let $a \in E - \{d\}$ be fixed. Since $d \in \mathrm{sp}(B_a) - B_a$, there exists a circuit C such that $C - B_a = \{d\}$. We have $(C - \{d\}) - \mathrm{sp}(A_a) \neq \emptyset$ (otherwise $C - \{d\}$ and hence d itself would be spanned by A_a, which we supposed not to be the case). Let $b \in (C - \{d\}) - \mathrm{sp}(A_a)$. It follows from the assumption that there exist $A_b, B_b \subseteq E$ such that $b, d \notin A_b \cup B_b$ and $\mathrm{sp}(A_b \cup \{d\}) = \mathrm{sp}(B_b)$.

We will now show that the sets $\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b)$ and $B_b \cup (A_a - A_b)$ and the branches d and b satisfy the conditions of Lemma 5. Clearly, $d \in (\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b)) - (B_b \cup (A_a - A_b))$ and $b \notin (\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b)) \cup (B_b \cup (A_a - A_b))$. To prove that $b \in \mathrm{sp}(\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b)) = \mathrm{sp}(B_b \cup (A_a - A_b))$, note that $\mathrm{sp}(\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b))$ contains

- $\{d\} \cup A_b$ and hence $\mathrm{sp}(\{d\} \cup A_b) = \mathrm{sp}(B_b)$ and B_b;
- both $(B_a - \{b\}) - B_b$ and $\mathrm{sp}(B_b) \supseteq B_b \supseteq (B_a - \{b\}) \cap B_b$, and hence $B_a - \{b\}$, $B_a \cup \{d\} - \{b\}$, $C - \{b\}$, $\mathrm{sp}(C - \{b\})$, $\{b\}$, B_a, $\mathrm{sp}(B_a) = \mathrm{sp}(A_a \cup \{d\})$ and A_a.

Thus, $\mathrm{sp}(\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b))$ contains $B_b \cup A_a \supseteq B_b \cup (A_a - A_b)$ and hence $\mathrm{sp}(B_b \cup (A_a - A_b))$. Similarly, $\mathrm{sp}(B_b \cup (A_a - A_b))$ contains

- B_b and hence $\mathrm{sp}(B_b) = \mathrm{sp}(A_b \cup \{d\})$ and $A_b \cup \{d\}$;



(a) $a \in A_3 - B_2$ (b) $a \in B_2 - A_2$ (c) $a \in A_2 - B_1$

(d) $a \in B_1 - A_1$ (e) $a \in A_1 - B_0$ (f) The result: A' (red) and B' (blue).

Figure 9: The process of substituting a for d, including the circuits C_b^A and C_b^B. At every step, the 'new' A and B are depicted. Edges in A are red, edges in B are blue, and circuits C_b^A and C_b^B are gray and dashed.

- both $A_a - A_b$ and $A_b \cup \{d\} \supseteq A_b \supseteq A_a \cap A_b$, and hence A_a, $A_a \cup \{d\}$, $\mathrm{sp}(A_a \cup \{d\}) = \mathrm{sp}(B_a)$ and B_a.

Thus, $\mathrm{sp}(B_b \cup (A_a - A_b))$ contains $A_b \cup \{d\} \cup B_a \supseteq \{d\} \cup A_b \cup ((B_a - \{b\}) - B_b)$ and hence $\mathrm{sp}(\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b))$. So $b \in \mathrm{sp}(\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b)) = \mathrm{sp}(B_b \cup (A_a - A_b))$.

Now, by Lemma 5, there exist $A', B' \subseteq E$ such that

- $A' \cup B' \subseteq (\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b)) \cup (B_b \cup (A_a - A_b)) \subseteq A_a \cup B_a \cup A_b \cup B_b - \{b\}$,
- $A' \cap B' \subseteq (\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b)) \cap (B_b \cup (A_a - A_b))$ and
- $\mathrm{sp}(A' \cup \{b\}) = \mathrm{sp}(B')$.

Let $A = A' \cup (B_a - B')$ and $B = B' \cup (A_a - A')$. To prove that $d \in \mathrm{sp}(A) = \mathrm{sp}(B)$, note that $\mathrm{sp}(A)$ contains

- A', $\{b\}$ (which is in B_a but not in B') and hence $A' \cup \{b\}$, $\mathrm{sp}(A' \cup \{b\}) = \mathrm{sp}(B')$ and B';
- both $B_a - B'$ and $B' \supseteq B_a \cap B'$, and hence B_a, $\mathrm{sp}(B_a) = \mathrm{sp}(A_a \cup \{d\})$ and A_a.

Thus, $\mathrm{sp}(A)$ contains $B' \cup A_a \supseteq B' \cup (A_a - A') = B$ and hence $\mathrm{sp}(B)$. Similarly, $\mathrm{sp}(B)$

contains

- B', $\mathrm{sp}(B') = \mathrm{sp}(A' \cup \{b\})$ and hence A';
- both $A_a - A'$ and $A' \supseteq A_a \cap A'$, and hence A', $A' \cup \{b\}$ and $\mathrm{sp}(A_a \cup \{b\})$. Since $b \in \mathrm{sp}(A_a \cup \{d\}) - \mathrm{sp}(A_a)$ and $b \notin A_a \cup \{d\}$, there exists a circuit C' such that $b \in C' \subseteq A_a \cup \{d\} \cup \{b\}$. We also have $d \in C'$ (otherwise $b \in \mathrm{sp}(A_a)$), so $d \in \mathrm{sp}(A_a \cup \{b\})$ and thus $\mathrm{sp}(A_a \cup \{b\}) = \mathrm{sp}(A_a \cup \{d\})$. Hence, $\mathrm{sp}(B)$ contains $\mathrm{sp}(A_a \cup \{d\}) = \mathrm{sp}(B_a)$ and B_a.

Thus, $\mathrm{sp}(B)$ contains $A' \cup B_a \supseteq A' \cup (B_a - B') = A$ and hence $\mathrm{sp}(A)$. We then have $d \in \mathrm{sp}(A) = \mathrm{sp}(B)$ and, since $d \notin A_a \cup B_a$, also $d \notin A \cup B$. Finally,

$$\begin{aligned} A \cap B &= (A' \cup (B_a - B')) \cap (B' \cup (A_a - A')) \\ &= (A' \cap B') \cup (A' \cap (A_a - A')) \cup ((B_a - B') \cap B') \cup ((B_a - B') \cap (A_a - A')) \\ &\subseteq (A' \cap B') \cup (B_a \cap A_a), \end{aligned}$$

and

$$\begin{aligned} A' \cap B' &\subseteq (\{d\} \cup A_b \cup ((B_a - \{b\}) - B_b)) \cap (B_b \cup (A_a - A_b)) \\ &= ((\{d\} \cup A_b) \cap B_b) \cup ((\{d\} \cup A_b) \cap (A_a - A_b)) \\ &\quad \cup (((B_a - \{b\}) - B_b) \cap B_b) \cup (((B_a - \{b\}) - B_b) \cap (A_a - A_b)) \\ &\subseteq (A_b \cap B_b) \cup (A_a \cap B_a). \end{aligned}$$

Hence, $A \cap B \subseteq (A_a \cap B_a) \cup (A_b \cap B_b) \subseteq \bigcup_{a \in E - \{d\}} (A_a \cap B_a)$. $\square$

5 Description of the Shannon Switching Game on a matroid

Now that we have developed some theory about matroids, we can describe how the Shannon Switching Game is to be played on such a matroid. The general situation from the graph version easily translates into this matroidal one: instead of a graph, we have a matroid and instead of edges, we have branches, one of which is distinguished and unplayable: e_*. Note that the vertices do not have a matroidal counterpart.[17] The goal of the Short player is to tag a subset $S \subseteq E$ that spans e_*. Since e_* itself is unplayable, this means that $S \cup e_*$ should contain a circuit and hence the Short player should aim to tag the branches $C - \{e_*\}$ of some circuit $C \in \mathfrak{C}$ with $e_* \in C$. Again, the Cut player's goal is to avoid that this happens.

The Shannon Switching Game on a matroid is either short, cut or neutral, following the same reasoning as for the graphical version. Indeed, if we let M be the graphical matroid on a given graph G, the classes of the Shannon Switching Games played on M and G correspond.

[17] This is the reason why it was necessary to distinguish one edge in the graphical version.

6 A winning strategy for the Short player

We now present Lehman's theorem (Theorem 14 in [12]) about a set of necessary and sufficient conditions for a Shannon Switching Game to be short, whose proof suggests a strategy for the Short player.[18]

Theorem 1. *Consider the Shannon Switching Game played on a matroid M with distinguished branch e_*. It is short if and only if there exist $A, B \subseteq E$ such that*

(i) $e_* \notin A \cup B$;

(ii) $A \cap B = \emptyset$;

(iii) $e_* \in \mathrm{sp}(A) = \mathrm{sp}(B)$.

Proof. We will first prove that the three conditions are sufficient for the Shannon Switching Game to be short: suppose that there exist $A, B \subseteq E$ such that $e_* \notin A \cup B$, $A \cap B = \emptyset$ and $e_* \in \mathrm{sp}(A) = \mathrm{sp}(B)$. Suppose also that the Cut player moves first. At turn i of the Cut player (that is, just before he or she moves), let $A_i \subseteq E$ denote the set of branches that are either still untagged and in A, or already tagged by the Short player (and thus in $A \cup B$, as we will see). That is

$$A_i := \{e \in A \mid e \text{ untagged before turn } i\} \cup \{e \in A \cup B \mid e \text{ tagged by Short before turn } i\}$$

and similarly

$$B_i := \{e \in B \mid e \text{ untagged before turn } i\} \cup \{e \in A \cup B \mid e \text{ tagged by Short before turn } i\}.$$

We hence have $A_1 = A$ and $B_1 = B$.

We will now show that the Short player can play in such a way that for every turn i, we have $\mathrm{sp}(A_i) = \mathrm{sp}(B_i) = \mathrm{sp}(A)$ and (hence) $e_* \in \mathrm{sp}(A_i)$. We do this by induction. For $i = 1$ this clearly holds, since $A_1 = A$ and $B_1 = B$. Now, let $N \geq 1$ and suppose we have $\mathrm{sp}(A_i) = \mathrm{sp}(B_i) = \mathrm{sp}(A)$ for all $1 \leq i \leq N$.

If $e_* \in \mathrm{sp}(A_N \cap B_N) = \mathrm{sp}(\{e \in A \cup B \mid e \text{ tagged by Short before turn } N\})$, the Short player has tagged a subset of the branches that spans e_* and has hence won.[19] If not, the Cut player tags some (untagged) branch $a \in E - (A_N \cap B_N)$, which set falls apart in $E - (A_N \cup B_N)$, $A_N - B_N$ and $B_N - A_N$.

- If $a \in E - (A_N \cup B_N)$, then there trivially exists a branch $b \in B_N - \mathrm{sp}(A_N \cap B_N)$ such that $\mathrm{sp}(A_{N+1}) = \mathrm{sp}(A_N \cup \{b\}) = \mathrm{sp}(B_N) = \mathrm{sp}(B_{N+1})$[20]: Short can tag any branch, since Cut 'broke' neither A_N nor B_N.

[18] This strategy makes use of two subsets of the branches (A and B), but Lehman does not say how to determine them. In Section 10 we cover a method for determining those subsets, although there, we consider only graphs.

[19] In this case, the players do not tag branches anymore, but we say they still 'take turns', so as to not make the induction more complex than necessary.

[20] Analogously, there exists a branch $b \in A_N - \mathrm{sp}(A_N \cap B_N)$ such that $\mathrm{sp}(B_{N+1}) = \mathrm{sp}(B_N \cup \{b\})$.

- If $a \in A_N - B_N$, then by Lemma 3 there exists a branch $b \in B_N - \mathrm{sp}(A_N \cap B_N)$ such that $\mathrm{sp}((A_N \cup \{b\}) - \{a\}) = \mathrm{sp}(B_N)$. Short should play on b, so that $\mathrm{sp}(A_{N+1}) = \mathrm{sp}((A_N \cup \{b\}) - \{a\}) = \mathrm{sp}(B_N) = \mathrm{sp}(B_{N+1})$.
- If $a \in B_N - A_N$, then by Lemma 3 there exists a branch $b \in A_N - \mathrm{sp}(A_N \cap B_N)$ such that $\mathrm{sp}((B_N) \cup \{b\}) - \{a\}) = \mathrm{sp}(A_N)$. Short should play on b, so that $\mathrm{sp}(B_{N+1}) = \mathrm{sp}((B_N \cup \{b\}) - \{a\}) = \mathrm{sp}(A_N) = \mathrm{sp}(A_{N+1})$.

In any of these cases, we have $e_* \in \mathrm{sp}(A_{N+1}) = \mathrm{sp}(B_{N+1}) = \mathrm{sp}(A)$. This completes the induction.

Since $e_* \in \mathrm{sp}(A_i) = \mathrm{sp}(B_i)$ for any i and after a finite number of turns n we have $A_n = \{e \in A \cup B \mid e \text{ tagged by Short before turn } n\}$, e_* will eventually be spanned by branches tagged by the Short player. Hence, the game is short.

We will now prove that the three conditions are necessary for the Shannon Switching Game to be short. Suppose it is short, and suppose first that $|E| \leq 3$. Then either $\{e_*\}$ itself is a circuit, or we have $E = \{e_*, a, b\}$ with $\mathfrak{C} = \{\{e_*, a\}, \{e_*, b\}, \{a, b\}\}$. (This becomes clear simply by checking all possibilities.) In the former case $A = B = \emptyset$ satisfy the three conditions; in the latter $A = \{a\}$ and $B = \{b\}$ do.

Now, let $N \geq 3$ and suppose the theorem is true for all matroids with $|E| \leq N$. Let $M = (E, \mathfrak{C})$ be a matroid with $|E| = N + 2$ such that the Shannon Switching Game played on M with respect to some branch e_* is short. If the Cut player plays first on a branch a, the Short player can respond by playing on a branch b (where $\{b\}$ is not a circuit[21]) such that the resulting, reduced game on $M^r = (E^r, \mathfrak{C}^r)$ is still winning for him. Here, $E^r = E - \{a, b\}$ and $\mathfrak{C}^r = \mathfrak{C}_1^r \cup \mathfrak{C}_2^r$, with $\mathfrak{C}_1^r = \{C - \{b\} \subseteq E \mid C \in \mathfrak{C}, a \notin C, b \in C\}$ and $\mathfrak{C}_2^r = \{C \in \mathfrak{C} \mid a, b \notin C, C^r \subsetneq C \text{ for all } C^r \in \mathfrak{C}_1^r\}$; or in words: the circuits C in $\mathfrak{C}$ that contained a are not circuits anymore. The circuits C that contained b (but not a) become circuits $C - \{b\}$. The circuits C that did not contain b (and a) stay circuits C, except when they now contain a (new) circuit $C - \{b\}$, since circuits cannot contain other circuits.

Since $|E^r| = |E| - 2 = N$, the theorem holds for this reduced game (by the induction hypothesis), so there exist $A_a^r, B_a^r \subseteq E^r$ such that $e_* \notin A_a^r \cup B_a^r$, $A_a^r \cap B_a^r = \emptyset$ and $e_* \in \mathrm{sp}(A_a^r) = \mathrm{sp}(B_a^r)$ (where the spans are taken with respect to M^r). It follows that, in the original matroid M, the subsets $A_a, B_a \subseteq E$ with $A_a = A_a^r \cup \{b\}$ and $B_a = B_a^r \cup \{b\}$ satisfy $a, e_* \notin A_a \cup B_a$, $A_a \cap B_a = \{b\}$ and $e_* \in \mathrm{sp}(A_a) = \mathrm{sp}(B_a)$ (where the spans are taken with respect to M). By Lemma 4, there exist $A'_a, B'_a \subseteq E$ such that $A'_a \cup B'_a \subseteq A_a \cup B_a$ (and hence $a, e_* \notin A'_a \cup B'_a$), $A'_a \cap B'_a \subseteq A_a \cap B_a - \{b\} = \emptyset$ and $\mathrm{sp}(A'_a \cup \{e_*\}) = \mathrm{sp}(B'_a)$. Clearly, $E - \{e_*\} \neq \emptyset$, and, since Cut could have played on any branch except e_*, this holds for all $a \in E - \{e_*\}$. Now by Lemma 6 there exist $A, B \subseteq E$ such that $e_* \notin A \cup B$, $A \cup B \subseteq \bigcup_{a \in E - \{e_*\}} (A'_a \cap B'_a) = \emptyset$ and $e_* \in \mathrm{sp}(A) = \mathrm{sp}(B)$. This completes the induction. □

The strategy that Short should follow is thus as follows: if Cut plays on a branch in A_i, Short should play on a branch in B_i so that $\mathrm{sp}(A_{i+1}) = \mathrm{sp}(A_i)$. If Cut plays on branch in B_i,

[21] Demanding this makes the proof a little easier.

Short should play on a branch in A_i so that $\text{sp}(B_{i+1}) = \text{sp}(B_i)$. If Cut plays on $E-(A_i \cup B_i)$, Short can play on any (unplayed) branch in $A_i \cup B_i$.[22]

6.1 Short's strategy on a graph

In the Shannon Switching Game played on a graph, this strategy is quite intuitive. If the game is short, there exist two subsets of the edges A and B that

- do not contain e_*,
- have an empty intersection and
- span the same subset of the edges, which contains e_*.

We can restrict A and B to spanning trees A^t and B^t of the subgraph corresponding to the edges in $\text{sp}(A) = \text{sp}(B)$ by leaving out edges that do not add to the span; it is easy to see that A^t and B^t still satisfy the above properties. We now have two disjoint trees that span both e_* and each other. That means that there exist two paths between u and v (recall that $e_* = (u, v)$), one consisting of edges in A^t and one of edges in B^t. If the Cut player now deletes (tags) an edge a of one of these paths (and thereby 'breaks' it), say the one consisting of edges in A^t, not all vertices incident with the edges in $\text{sp}(A^t)$ are connected anymore by edges in $A^t - \{a\}$. Since they are still connected by edges in B^t, Short can tag an edge b in B^t such that $A^t - \{a\} \cup \{b\}$ is again a spanning tree of the subgraph corresponding to the edges in $\text{sp}(A^t)$. Hence, we again have two sets (i.e. $A^t - \{a\} \cup \{b\}$ and B^t) that do not contain e_* and span the same subset, which contains e_*. They are, however, not disjoint anymore; this is okay because Cut cannot tag their common edge b (and 'break' both paths at once), since it has already been tagged.

An example of this strategy is given in Figure 10. It is easily verified that A and B (which, in this case, are already trees themselves) satisfy the three conditions. We illustrate some of the moves. In Figure 10b, the Cut player breaks (the vertices incident with edges in) B up into two parts; Short remedies this by tagging an edge in A that reconnects these two parts again. In Figure 10k, Short could already have won by tagging one of the two upper bent edges, but the strategy tells him to do otherwise. After Cut plays outside $A \cup B$ in Figure 10l, Short can play on any edge.[23] Finally, in Figure 10o, Short completes a cycle that contains e_* and hence wins.

[22] Actually, Short can play on any (unplayed) branch in E; however, if Short plays on a branch in $E - (A_i \cup B_i)$, we do not have $\text{sp}(A_i) = \text{sp}(A)$ anymore, which would make the proof a bit more complicated.

[23] Note that this play by Short (see Figure 10m) causes B_i to contain a cycle and hence cease to be a tree. This can be remedied by deleting an untagged edge in the cycle from B, or we can simply ignore it and let Short play on any edge if Cut plays on an edge from the cycle.

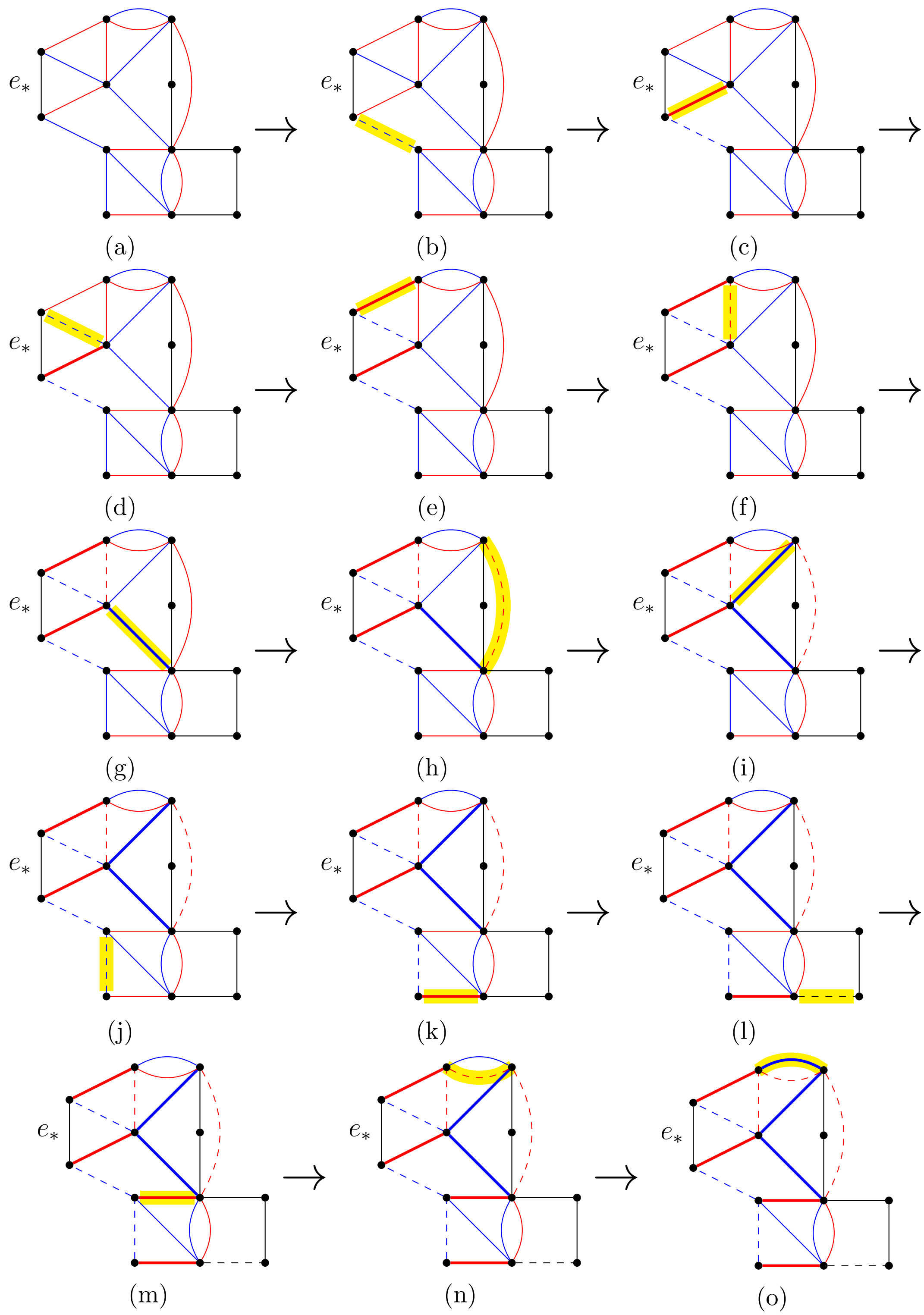


Figure 10: An example of the Short player's strategy. Edges in A are red, edges in B are blue, edges tagged by Short are thick and edges tagged by Cut are dashed. Hence, at turn i, A_i consists of the thick edges plus the red edges that are neither thick nor dashed. B_i consists of the thick edges plus the blue edges that are neither thick nor dashed. The edge last played on is highlighted yellow.

7 Dual matroids

Now that we have presented a winning strategy for the Short player (when the game is short), we would like to do the same for the Cut player (when the game is cut). As it turns out, the goals of the two players are, in a way, similar to each other.[24] Because of this, we do not need to find a completely new strategy for the Cut player; instead, we can translate Short's strategy into one for Cut. To do so, we first need the concept of the dual of a matroid.

Like a matroid itself, the dual of a matroid can be defined in different (equivalent) ways. For the same reason as before, we do this in terms of circuits.

Let $M = (E, \mathfrak{C})$ be a matroid. Then $M' = (E, \mathfrak{C}')$ is called the *dual matroid*, or simply the *dual*, of M, where $\mathfrak{C}' \subseteq 2^E$ is the collection of subsets of E respecting the following properties:

1. $\emptyset \notin \mathfrak{C}'$.
2. If $C'_1, C'_2 \in \mathfrak{C}'$ and $C'_1 \subseteq C'_2$, then we have $C'_1 = C'_2$.
3. $|C' \cap C| \neq 1$ for all $C' \in \mathfrak{C}'$ and $C \in \mathfrak{C}$.

We denote the span of a subset $A \subseteq E$ with respect to $\mathfrak{C}'$ by $\mathrm{sp}'(A)$, that is

$$\mathrm{sp}'(A) := \{e \in E \mid e \in A \text{ or there exists a circuit } C' \in \mathfrak{C}' \text{ such that } e \in C' \subseteq A \cup \{e\}\}.$$

Lehman [12, p. 701] offers the following alternative definition (here presented as a lemma) of the dual matroid, which is used in the proof of Lemma 8. We refer to [12] for the proof of Lemma 7.

Lemma 7. *Let $e \in A \in E$. Then $A \in \mathfrak{C}'$ if and only if A is a minimal set such that $A \cap C - \{e\} \neq \emptyset$ for all C such that $e \in C \in \mathfrak{C}$.*

Although there does not exist a (universally accepted) general definition of duality, the definition of the dual matroid above satisfies two of its common properties. The first is that the dual of a matroid is again a matroid. The second is that the dual of the dual of a matroid yields the original matroid: $M'' = M$. For the proofs of both properties, see for example [12] (in terms of circuits)[25] or [9] (in terms of bases).

8 A winning strategy for the Cut player

Now that we have defined the dual of a matroid and considered some of its properties, we can make the way in which the goals of the two players are similar to each other precise. (In [12], this is Lemma 21.)

Lemma 8. *The Short player wins the game on M if and only if the Cut player wins the game on M'.*

[24] This similarity becomes immediately clear in a specific version of the Shannon Switching Game, which is known as Gale or Bridg-It. See, for example, [6, p. 84] and [12, p. 714].

[25] In the second line of the proof of (19) from [12] there is a typo: $A, B \in \mathfrak{M}$ should be $A, B \in \mathfrak{M}'$.

Proof. Suppose first that the Short player wins the game on M. That means that there is a circuit $C \in \mathfrak{C}$ with $e_* \in C$ where every branch in $C - \{e_*\}$ is tagged by the Short player. Thus, in the game on M', these branches are tagged by the Cut player. From (the dual version of) Lemma 7 it now follows that $C \cap C' - \{e_*\} \neq \emptyset$ for all C' such that $e_* \in C' \in \mathfrak{C}'$, and hence every circuit in M' that contains e_*, contains a branch tagged by the Cut player. Hence, the Cut player wins the game on M'.

Suppose now that the Cut player wins the game on M'. That means that every circuit $C' \in \mathfrak{C}'$ with $e_* \in C'$ contains (at least) one branch that is tagged by the Cut player, say $a_{C'}$. Let $A = \{a_{C'} \mid e_* \in C' \in \mathfrak{C}'\}$. Then $A \cap C' - \{e_*\} \neq \emptyset$ for all C' such that $e_* \in C' \in \mathfrak{C}'$. Now, let $A^{min} \subseteq A$ be minimal such that $A^{min} \cap C' - \{e_*\} \neq \emptyset$ for all C' such that $e_* \in C' \in \mathfrak{C}'$. Then by Lemma 7, we have $A^{min} \in \mathfrak{C}'' = \mathfrak{C}$. Since every branch in $A - \{e_*\}$ (and hence every branch in $A^{min} - \{e_*\}$) is tagged by the Cut player in the game on M', every branch in $A^{min} - \{e_*\}$ is tagged by the Short player in the game on M. Since A^{min} is a circuit in M, the Short player wins the game on M. □

Lemma 8 allows us to 'translate' the conditions for a game to be short (given in Theorem 1) to conditions for a game to be cut, for which only the primal game needs to be considered. (In [12], this is Theorem 26.)

Theorem 2. *Consider the Shannon Switching Game played on a matroid M with distinguished branch e_*. It is cut if and only if there exist $A, B \subseteq E$ such that*

(i) $e_* \notin A \cup B$;

(ii) $A \cap B = \emptyset$;

(iii) $e_* \in C \in \mathfrak{C}$ *implies* $A \cap C \neq \emptyset$ *(and* $B \cap C \neq \emptyset$*)*;

(iv) $C \in \mathfrak{C}$ *implies* $A \cap C \neq \emptyset$ *if and only if* $B \cap C \neq \emptyset$.

Proof. Combining Theorem 1 and Lemma 8, we get that the game is cut if and only if there exist $A, B \subseteq E$ such that $e_* \notin A \cup B$, $A \cap B = \emptyset$ and $e_* \in \mathrm{sp}'(A) = \mathrm{sp}'(B)$.

Suppose first that the game is cut. From the above it is clear that we only have to prove the third and fourth property of A and B, which we prove simultaneously: first, let $C \in \mathfrak{C}$ be a circuit such that $e_* \in C$ and/or $A \cap C \neq \emptyset$. Then there exists a branch $a \in C \cap (A \cup \{e_*\}) \subseteq \mathrm{sp}'(A) = \mathrm{sp}'(B)$, where the inclusion follows from the fact that both A and $\{e_*\}$ are spanned by A. Since $a \in A \cup \{e_*\}$, $a \notin B$ and hence $a \in \mathrm{sp}'(B) - B$. This means that there exists a circuit $C' \in \mathfrak{C}'$ such that $a \in C' \subseteq B \cup \{a\}$ and hence $C \cap B \supseteq C \cap C' - \{a\} \neq \emptyset$, where the inequality follows from the third property of the definition of the dual. By symmetry, $B \cap C \neq \emptyset$ implies $A \cap C \neq \emptyset$.

Suppose now that the four properties of A and B hold. For the same reason as above, we here only have to prove $e_* \in \mathrm{sp}'(A) = \mathrm{sp}'(B)$. Let $a \in A \cup \{e_*\}$ be a branch. If $a \in C \in \mathfrak{C}$, then we have $e_* \in C \in \mathfrak{C}$ and/or $A \cap C \neq \emptyset$, both of which imply $B \cap C \neq \emptyset$ by assumption. Note that $B \cap C = (B \cup \{a\}) \cap C - \{a\}$, which thus is not empty either. Now, let $B^{min} \subseteq B$ be a minimal set such that still $(B^{min} \cup \{a\}) \cap C - \{a\} \neq \emptyset$, then by Lemma 7 we have $B^{min} \cup \{a\} \in \mathfrak{C}'$. Since $a \in B^{min} \cup \{a\} \subseteq B \cup \{a\}$, a is spanned by B with respect to $\mathfrak{C}'$ and hence $A \cup \{e_*\} \subseteq \mathrm{sp}'(B)$. By symmetry, we also have $B \cup \{e_*\} \subseteq \mathrm{sp}'(A)$.[26] From these two inclusions it follows that we have $e_* \in \mathrm{sp}'(A) = \mathrm{sp}'(B)$ and hence the game is cut. □

The strategy that Cut should follow, is found by dualizing the strategy for Short in the proof of Theorem 1. If we set

$$A'_i := \{e \in A \mid e \text{ untagged before turn } i\} \cup \{e \in A \cup B \mid e \text{ tagged by Cut before turn } i\}$$

and

$$B'_i := \{e \in B \mid e \text{ untagged before turn } i\} \cup \{e \in A \cup B \mid e \text{ tagged by Cut before turn } i\},$$

then Cut should play as follows: if Short plays on a branch in A'_i, Cut should play on a branch in B'_i so that $\mathrm{sp}'(A'_{i+1}) = \mathrm{sp}'(A'_i)$. If Short plays on branch in B'_i, Cut should play on a branch in A'_i so that $\mathrm{sp}'(B'_{i+1}) = \mathrm{sp}'(B'_i)$. If Short plays on $E - (A'_i \cup B'_i)$, Cut can play on any (unplayed) branch in $A'_i \cup B'_i$ (or even in E).

8.1 Cut's strategy on a graph

As for Short's strategy, we now give a more intuitive strategy for the Cut player for the Shannon Switching Game played on a graph. If the game is cut, there exist two subsets of the edges A and B that

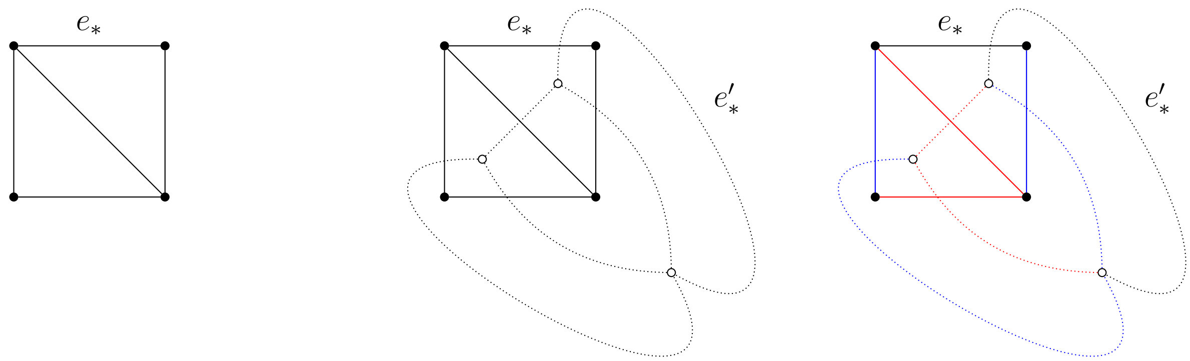
- do not contain e_*,
- have an empty intersection and
- all circuits that contain an edge in one of the subsets, also contain an edge in the other.
- Furthermore, all circuits that contain e_*, also contain an edge in one subset (and hence also an edge in the other).

That means that every path between u and v contains at least one edge from both A and B. Suppose the Short player tags an edge a so that there now exists a circuit that does not contain an untagged edge anymore in one of the two sets (but previously did), say A. (Note that there is at most one such circuit: suppose there were two, then the circuit consisting of these two circuits minus the tagged edge would not have contained edges in A in the first place.) Since the circuit still contains an untagged edge in B, Cut can tag an edge b in B such that all circuits (that previously contained edges in both A and B) again contain edges in both $A - \{a\} \cup \{b\}$ and B. Hence, we again have two sets (i.e. $A - \{a\} \cup \{b\}$ and B) that satisfy the above properties, except that they are not disjoint anymore; this is okay because Short cannot tag their common edge b (and thereby tag edges from both sets in a circuit at once), since it has already been tagged.

Alternatively, Cut's strategy for a Shannon Switching game played on a graph can be found by following Short's strategy on the dual graph (due to Lemma 8). Note that the dual graph only exists if the (primal) graph is planar.[27] A simple example of this procedure is depicted in

[26] In [12] the prime symbol (′) is missing here.
[27] See, for example, [9, p. 60].

Figure 11. In Figure 11b the dual graph is found by placing a vertex in every face (including 'the outside'), and for every edge in the primal graph, placing an edge that connects the vertices corresponding to the faces on both sides of that primal edge. In Figure 11c A and B are indicated in both graphs, satisfying the conditions of Theorem 1 in the dual graph, and the conditions of Theorem 2 in the primal graph. Cut can now win the game on the primal graph by following Short's strategy on the dual, where the intersecting edges correspond with each other.[28]



(a) A graph. (b) Its dual graph (dotted). (c) A (red) and B (blue).

Figure 11: An example of the procedure of finding Cut's strategy by dualizing the graph and playing Short's strategy on it.

If the graph is not planar, we have to follow the 'matroidal' strategy given in Theorem 2. An example of such a graph is $K_{3,3}$, which indeed renders a cut game (irrespective of the chosen edge e_*). Note that K_5, the other 'fundamental nonplanar graph', renders a short game.

An example of this strategy is given in Figure 12. It is easily verified that A and B satisfy the four conditions. We illustrate some of the moves. In Figure 12b, the Short player tags an edge a in A so that the cycle $\{e_*, a, b\}$ does not contain an edge in A anymore that is or can be deleted by Cut. Cut remedies this by tagging b (and 'adding' it to A). After Short's play in Figure 12l, Cut could already have won by playing on d, but the strategy demands a play on f to ensure that the cycle $\{f, g, h\}$ once again contains an edge in both sets. Finally, in Figure 12o, Cut tags an edge that makes sure that Short cannot win anymore and hence Cut wins.

9 Winning strategies in neutral games

We have seen how Short can win if the game is short, and how Cut can win if the game is cut. We will now see how both players can win in a neutral game if they play first.

In a graph, we can turn a neutral game into a short game by adding an edge with end points u and v (where still $e_* = (u, v)$) [12]. (See Figure 13b.) To see that this game is indeed

[28]Technically, we have drawn *a* dual graph, although we can speak of *the* (abstract) dual if we do not draw it.

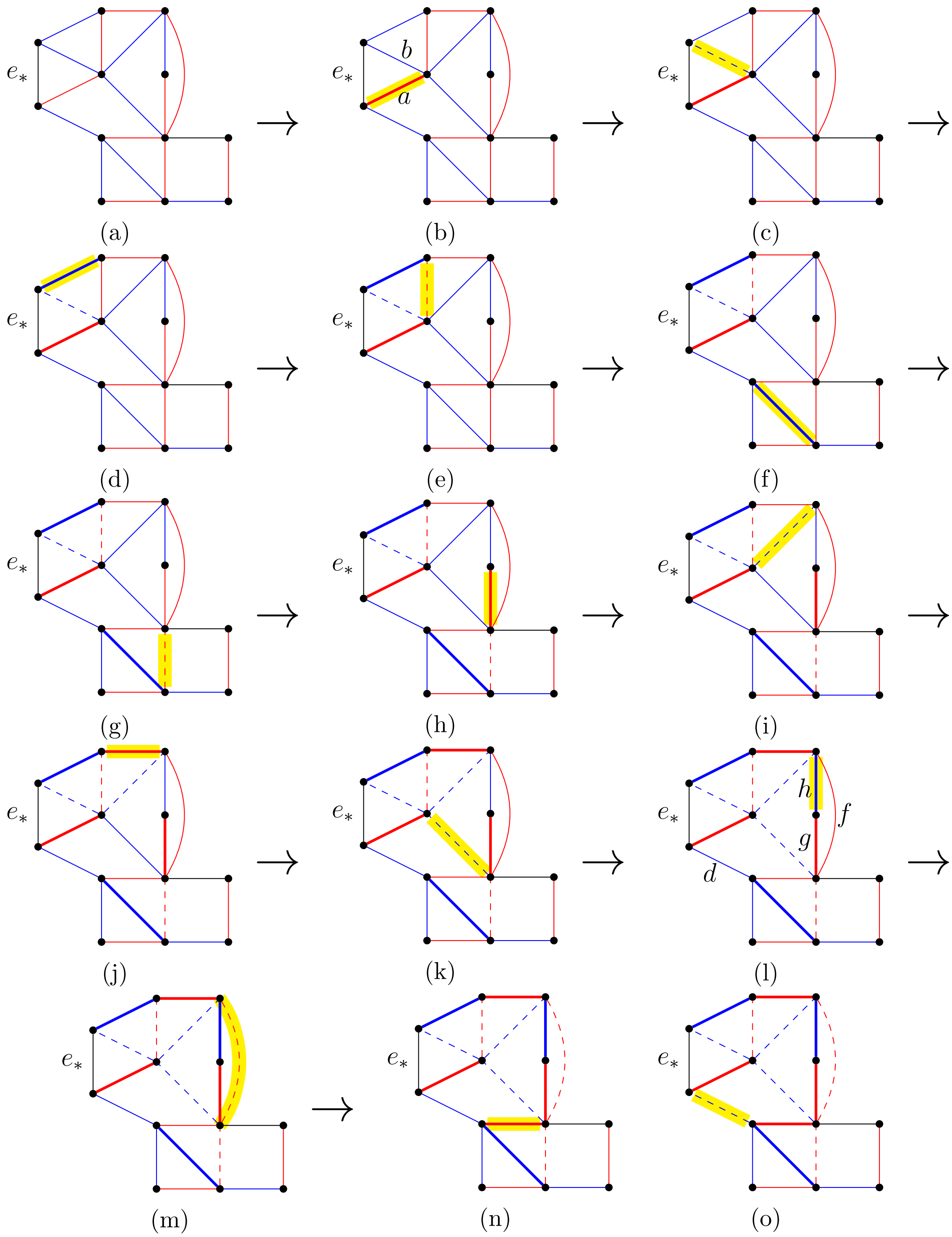
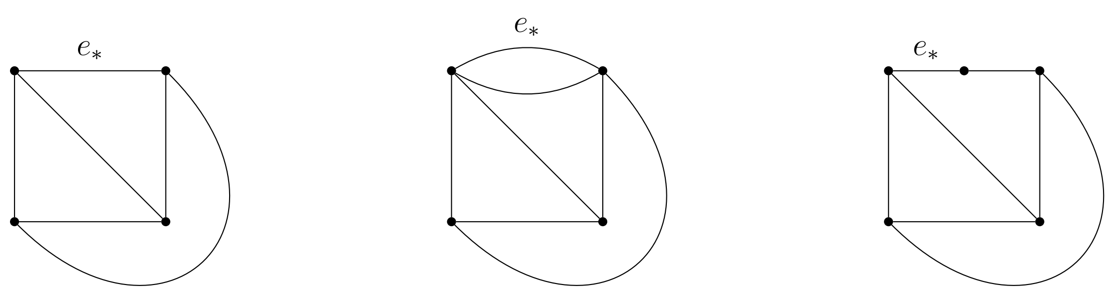


Figure 12: An example of the Cut player's strategy. Edges in A are red, edges in B are blue, edges tagged by Short are thick and edges tagged by Cut are dashed. Hence, at turn i, A'_i consists of the dashed edges plus the red edges that are neither thick nor dashed. B'_i consists of the thick edges plus the blue edges that are neither thick nor dashed. The edge last played on is highlighted yellow.

short, note that if Cut does not play on the added edge, Short can win by playing on it; if Cut does play on the added edge, the resulting game is the neutral game that we started with, and Short can play first. A winning strategy for the Short player in the neutral game can now be found by following the strategy for the short game after Cut would have played on the added edge.

On a matroid, we can perform this same procedure by adding a branch a to E and adding the circuit $\{e, a\}$ to $\mathfrak{C}$, as well as circuits $C - \{e\} \cup \{a\}$ for every circuit C that contains e.

We can now find a winning strategy for the Cut player in a neutral game on a graph in two ways. We can either dualize the short game that was used to find a winning strategy for Short (where we now suppose that Short plays first on the added edge), or we can turn the neutral game directly into a cut game by adding a vertex to the edge e_*, thereby 'splitting' it into two edges and setting one of these two edges to be the new distinguished edge e_*. (See Figure 13c.) On a matroid, this corresponds to adding a branch a to E and replacing all circuits C containing e_* by $C \cup \{a\}$.



(a) A neutral game.

(b) The neutral game made short.

(c) The neutral game made cut.

Figure 13: Turning a neutral game into either a short or a cut game.

Note that we can now also check whether a game is neutral by adding an edge with end points u and v and then checking if the game is short. If that newly obtained game is short, but the original game was not, it must have been neutral.[29]

10 Finding a pair of disjoint cospanning sets

As Bruno and Weinberg [2] note, Lehman's solution to the Shannon Switching Game 'is not fully satisfactory. The basic theorem on so-called short games is given in terms of the existence of a pair of disjoint cospanning[30] sets, but it appears in nonconstructive form; that is, no method is given for generating such a pair or even determining whether they exist in a given graph or matroid.' Bruno and Weinberg then combine Lehman's results with those

[29] A similar procedure where a vertex is added to e_* and it is checked whether the game is cut also works.
[30] Two subsets $A, B \subseteq E$ are said to be *cospanning* if $\mathrm{sp}(A) = \mathrm{sp}(B)$.

of Kishi and Kajitani [10] to give a complete (graph-theoretic) solution of the game. For the determination of the cospanning sets, they refer to Bruno's Ph.D. dissertation [3]; the algorithms given there, however, are in terms of matroids.

In this section, we present a graph-theoretic procedure for determining a pair of subsets of the edges of a graph that satisfies the conditions for the Shannon Switching Game played on that graph to be short, as given in Theorem 1. (That is, if the game is a short game.) We have established this procedure by applying the procedure given in [10, pp. 325-326] to an instance of the Shannon Switching Game, the main difference being that the subsets may now not contain the distinguished edge e_*. A pair of subsets that satisfies the conditions for the game on that graph to be cut (if the game is a cut game) as given in Theorem 2 may be found by first constructing the dual graph (or matroid), then applying the procedure given below, and finally translating back the obtained subsets to the primal situation.

Given a graph G and a distinguished edge $e_* \in E(G)$, our goal is to determine two subsets of the edges $A, B \subseteq E(G)$ so that $e_* \notin A \cup B$, $A \cap B = \emptyset$ and $e_* \in \mathrm{sp}(A) = \mathrm{sp}(B)$.[31] We achieve this in three steps:

1. We determine two spanning trees of G.
2. We perform the procedure by Kishi and Kajitani to find a pair of maximally distant trees. (We will define (maximal) distance later.)
3. We make sure that neither tree contains e_* and find disjoint subtrees.

10.1 Step 1 - Finding two spanning trees

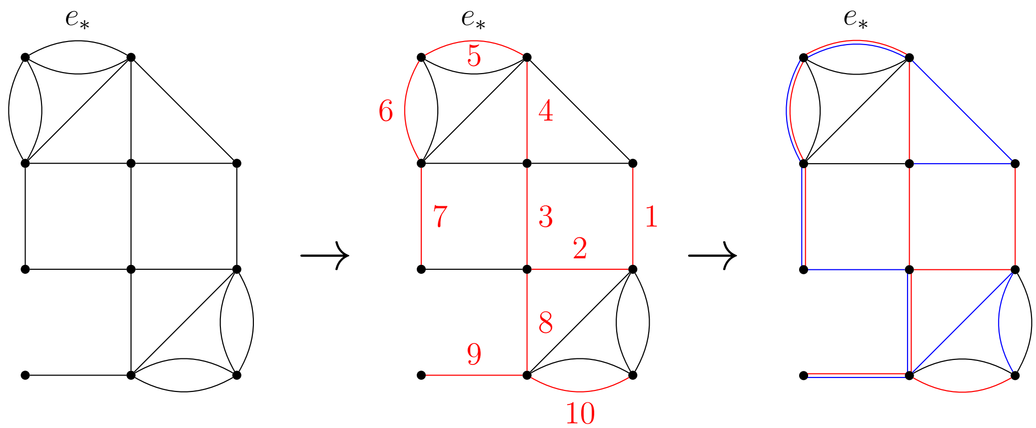
To determine two spanning trees, we can simply apply a tree finding algorithm like depth-first search or breadth-first search.[32] For example, depth-first search tells you to start at some vertex of the graph and then walk along an edge to an adjacent vertex. If you have not visited that vertex before, you add the edge to some subset $T \subseteq E(G)$ (which is initialized as $\emptyset$), walk on to a vertex adjacent to the last one, and continue depending on whether you have visited that vertex already.

If you have visited a vertex before, you do not add the edge to T but instead walk back along the same edge and try another adjacent vertex. When you have visited all adjacent vertices, you go back one more vertex and try the vertices adjacent to that one. When you have visited all vertices, you stop. T is now a spanning tree of G: it connects all vertices and contains no cycles.

To obtain a second spanning tree, simply perform the algorithm again. Although the two trees are allowed to be equal, step 2 will generally be faster if the two trees are already more distant. Hence, choosing a different starting vertex may be helpful. Let (T_1, T_2) be the obtained pair of spanning trees. See Figure 14 for an example.

[31] In this section, $\mathrm{sp}(S)$ denotes the graphical span of a subset of the edges S, that is, $\mathrm{sp}(S) := \{e \in S \mid e \in S \text{ or there exists a cycle } C \text{ such that } e \in C \subseteq S \cup \{e\}\}$. Note that this definition agrees very naturally with that of the matroidal span.

[32] See, for example, [5].



(a) A graph and a distinguished edge e_*.

(b) The construction of T_1.

(c) The obtained pair of spanning trees (T_1, T_2).

Figure 14: Step 1. The construction of a pair of spanning trees using depth-first search. Edges in T_1 are red and the numbers in (b) indicate the order in which the algorithm adds the edges. Edges in T_2 are blue.

10.2 Step 2 - Maximally distant trees

Given a graph G, the *distance* $d(T_1, T_2)$ between two spanning trees T_1 and T_2 of G is defined to be the number of edges contained in one tree, but not in the other. Note that this is well-defined, since every spanning tree of the same graph contains the same number of edges (i.e. the number of vertices minus 1). A pair of spanning trees (T_1, T_2) is called *maximally distant* if there exists no pair of spanning trees (T_1', T_2') such that $d(T_1', T_2') > d(T_1, T_2)$.

Kishi and Kajitani offer a procedure to construct a pair of maximally distant spanning trees from an arbitrary pair of spanning trees. Before we give it here, we first need a few more definitions.

Given a graph G and a spanning tree T, a *chord* of T is an edge in $E(G) - T$. An edge that is a chord of two spanning trees T_1 and T_2 is called a *common chord* of T_1 and T_2. The *fundamental cycle* of a chord e with respect to T is the unique cycle in $T \cup \{e\}$.

Consider now again the pair of spanning trees (T_1, T_2) from step 1. If they have no common chords, they are maximally distant, since they overlap as little as possible. Suppose therefore that the set of common chords of (T_1, T_2) is not empty and let e be a common chord. Let L_e^1 be the fundamental cycle of e with respect to T_1. If $L_e^1 \cap T_2 \neq \emptyset$, then we can 'interchange' e and some $a \in L_e^1 \cap T_2$ to obtain a new spanning tree $T_1' = T_1 \cup \{e\} - \{a\}$. ($T_1'$ is still a spanning tree, since it connects all vertices and contains no cycles: if it did, T_1 would have contained a cycle as well.) Clearly we have $d(T_1', T_2) = d(T_1, T_2) + 1$.

If $L_e^1 \cap T_2 = \emptyset$, then we consider the second 'layer', L_e^2, which is the union of all fundamental cycles of the edges in L_e^1 with respect to T_2. If L_e^2 contains an edge in $T_1 \cap T_2$, say b, in the fundamental cycle of some edge $a \in L_e^1$ (with respect to T_2), then we can obtain two new spanning trees $T_1' = T_1 \cup \{e\} - \{a\}$ and $T_2' = T_2 \cup \{a\} - \{b\}$. Note that if b is in the fundamental cycle of e itself, b still has to be in the fundamental cycle of some edge $a \in L_e^1$ with $a \neq e$: suppose not, then e would be spanned by both its fundamental cycle and the union of the fundamental cycles of the edges in L_e^1, and hence T_2 would contain a cycle.[33]

To ensure that the process is clear, we also discuss what to do when $L_e^2 \cap (T_1 \cap T_2) = \emptyset$. L_e^3 then is the union of all fundamental cycles of the edges in L_e^2 with respect to T_1. If L_e^3 contains an edge in $T_1 \cap T_2$, say c, in the fundamental cycle of some edge $b \in L_e^2$ (with respect to T_1), then we can obtain two new spanning trees $T_1' = T_1 \cup \{e, b\} - \{a, c\}$ and $T_2' = T_2 \cup \{a\} - \{b\}$. If not, we continue with L_e^4.

If $L_e^i \cap (T_1 \cap T_2) = \emptyset$ for all i, we will get $L_e^k = L_e^{k-1} := L_e^\infty$ for some k: we have $L_e^i \subseteq L_e^{i-1}$ for all $i > 1$ and the number of edges is finite. Either when we have $L_e^k \cap (T_1 \cap T_2) \neq \emptyset$ for some k (and perform the interchanging procedure) or when we have $L_e^k = L_e^{k-1}$ for some k, we stop, and we start the process with another common chord, using the newly obtained trees if we have constructed them. We do this for every common chord. The pair of spanning trees that we end up with, say (T_1^*, T_2^*), is maximally distant. We refer to [10] for the proof. See Figure 15 for an example.

10.3 Step 3 - Finding disjoint subtrees that do not contain the distinguished edge

Steps 1 and 2 can be performed on any graph, even if the game played on that graph is not short. Note that we already have $e_* \in \mathrm{sp}(T_1^*) = \mathrm{sp}(T_2^*)$. If the game is short, we can now quite easily obtain two trees that both do not contain e_* (if that is not already the case) in the following way: there exists a (not necessarily unique) common chord e of T_1^* and T_2^* so that e_* is in some layer L_e^i around e.[34] We now perform the same procedure on e as we did in step 2, except that we try to find e_* instead of some edge in $T_1 \cap T_2$. When we find e_*, we perform the same interchanging procedure to obtain two spanning trees T_1^{**} and T_2^{**} that do not contain e_*.

To find a right common chord, we could just perform the described procedure on all common chords until e_* shows up in a layer around one of them. However, we could also already find one during the execution of step 2: if e_* is not a common chord itself, it must be in a layer around a common chord e whose layers do not contain an edge in $T_1 \cap T_2$ (that is, at the moment in the procedure that those layers are checked)[35] and hence is not interchanged. We can then perform the interchanging procedure starting at e.

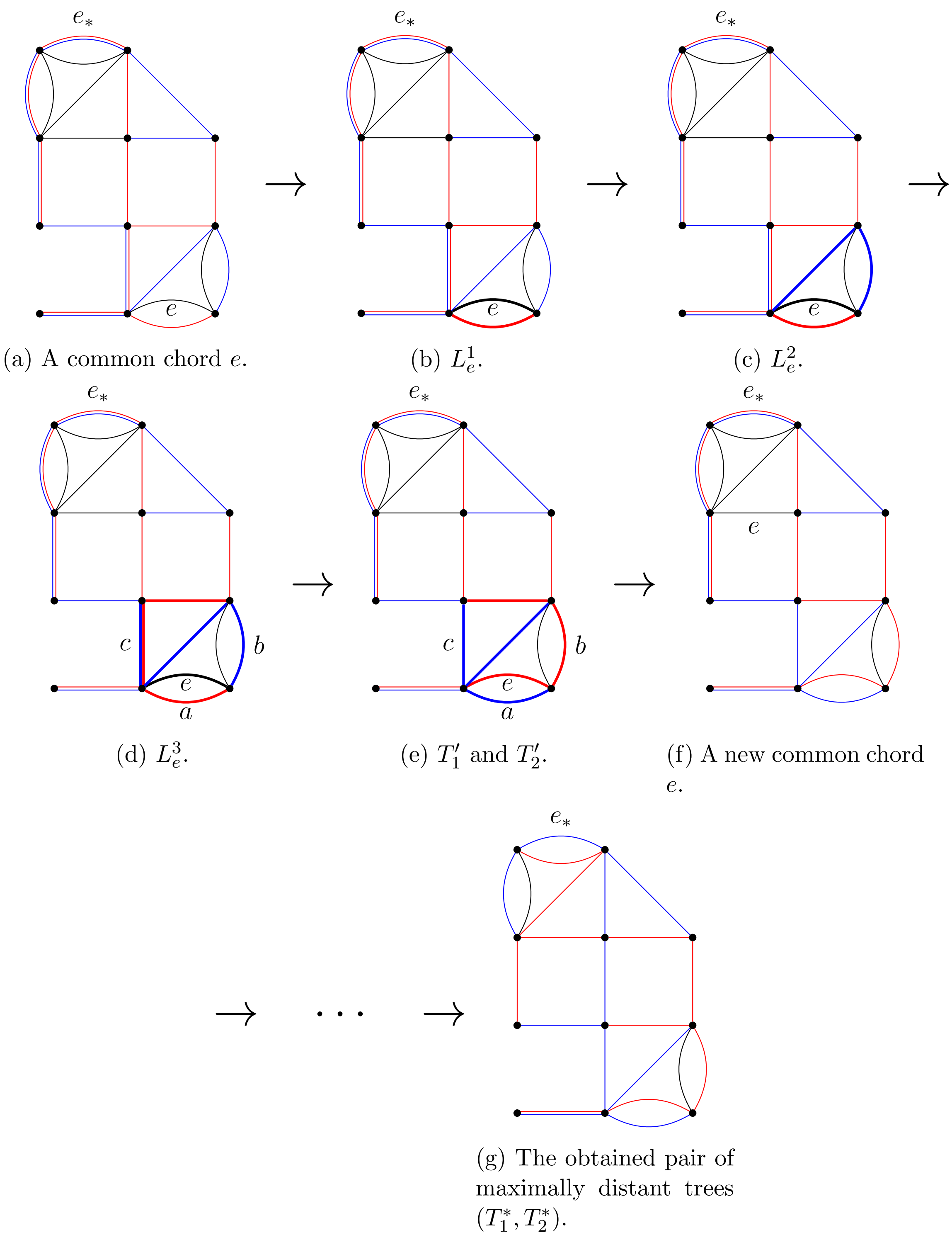
The former common chord e together with the layers around it (i.e. L_e^∞) or, equivalently, the current common chord e_* together with the layers around it (i.e. $L_{e_*}^\infty$), does, as mentioned, not contain an edge in $T_1^{**} \cap T_2^{**}$. Hence, the trees $T_1^{**} \cap L_{e_*}^\infty$ and $T_2^{**} \cap L_{e_*}^\infty$ are disjoint. The

[33] All fundamental cycles referred to in this last sentence are with respect to T_2.

[34] Kishi and Kajitani prove this and from this proof it is clear which common chord you should choose. For this proof, they develop some theory that goes beyond our purposes, so we omit it here. (To the reader who is familiar with the article: e_* should be in the so-called K subgraph of (T_1^*, T_2^*) with respect to e.) We will, however, cover two simple alternative methods of finding a right common chord.

[35] This again follows from the theory developed in [10]: e_* is in the K subgraph of (T_1^*, T_2^*) with respect to some common chord, that is, if the game with respect to e_* is indeed a short game.

other conditions clearly still hold, so we have found a pair of subsets of the edges of G that satisfies the conditions given in Theorem 1. See Figure 16 for an example.



(a) A common chord e.

(b) L_e^1.

(c) L_e^2.

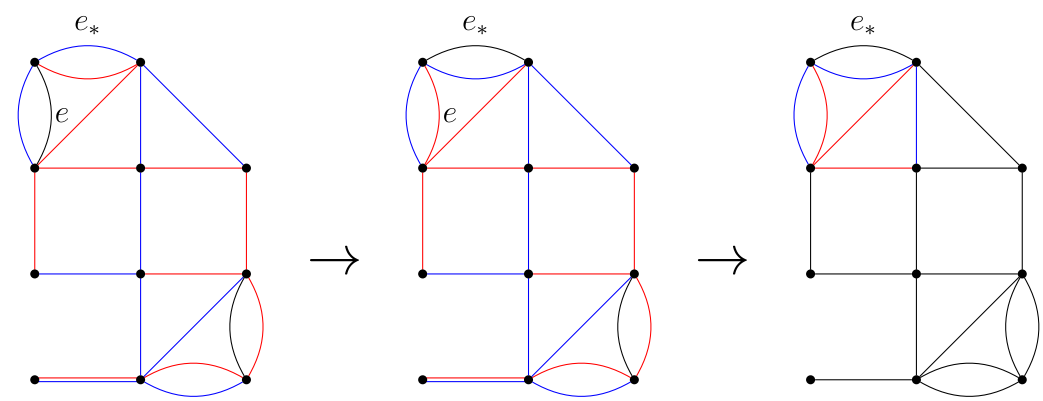
(d) L_e^3.

(e) T_1' and T_2'.

(f) A new common chord e.

(g) The obtained pair of maximally distant trees (T_1^*, T_2^*).

Figure 15: Step 2. The construction of a pair of trees (T_1', T_2') from (T_1, T_2), where $d(T_1', T_2') = d(T_1, T_2)+1$, and the final pair of maximally distant trees (T_1^*, T_2^*). Edges in T_1, T_1' and T_1^* are red, edges in T_2, T_2' and T_2^* are blue, and edges in L_e^i are thick. We have $T_1' = T_1 \cup \{e, b\} - \{a, c\}$ and $T_2' = T_2 \cup \{a\} - \{b\}$.



(a) A common chord e of (T_1^*, T_2^*) such that $e_* \in L_e^2$.

(b) Edges e and e_* are interchanged within T_1^* to obtain (T_1^{**}, T_2^{**}).

(c) The pair of disjoint (non-spanning) subtrees $T_1^{**} \cap L_{e_*}^{\infty}$ and $T_2^{**} \cap L_{e_*}^{\infty}$.

Figure 16: Step 3. The construction of two disjoint subtrees that do not contain the distinguished edge. Edges in T_1^*, T_1^{**} and $T_1^{**} \cap L_{e_*}^{\infty}$ are red, and edges in T_2^*, T_2^{**} and $T_2^{**} \cap L_{e_*}^{\infty}$ are blue.

References

[1] Michael Albert, Richard Nowakowski, and David Wolfe. *Lessons in Play: An Introduction to Combinatorial Game Theory, Second Edition*. 2nd. USA: A. K. Peters, Ltd., 2016. ISBN: 1482243032.

[2] J. Bruno and L. Weinberg. “A constructive graph-theoretic solution of the Shannon switching game”. In: *IEEE Transactions on Circuit Theory* 17.1 (1970), pp. 74–81. DOI: 10.1109/TCT.1970.1083056.

[3] John Louis Bruno. *Matroids, Graphs and Resistance Networks*. 1969.

[4] A. P. Cláudio et al. “Shannon Switching Game and Directed Variants”. In: *Dynamics, Games and Science*. Ed. by Jean-Pierre Bourguignon et al. Cham: Springer International Publishing, 2015, pp. 187–199. ISBN: 978-3-319-16118-1.

[5] Shimon Even. *Graph algorithms*. Computer software engineering series. London: Pitman, 1979. ISBN: 0914894218.

[6] M. Gardner. *The 2nd Scientific American Book of Mathematical Puzzles & Diversions: A New Selection: from Origami to Recreational Logic, from Digital Roots and Dudeney Puzzles to the Diabolic Square, from the Golden Ratio to the Generalized Ham Sandwich Theorem*. A fireside book. Simon & Schuster, 1961. URL: https://books.google.nl/books?id=Z9P8swEACAAJ.

[7] Yahya Ould Hamidoune and Michel Las Vergnas. “Directed Switching Games on graphs and matroids”. eng. In: *Journal of combinatorial theory. Series B* 40.3 (1986), pp. 237–269. ISSN: 0095-8956.

[8] John Horgan. *Claude Shannon: Tinkerer, Prankster, and Father of Information Theory*. 2016. URL: https://spectrum.ieee.org/claude-shannon-tinkerer-prankster-and-father-of-information-theory (visited on 02/07/2024).

[9] L.C.M Kallenberg and M. Schouten-Straatman. *Geavanceerde onderwerpen in de besliskunde*. July 2020.

[10] G. Kishi and Y. Kajitani. “Maximally Distant Trees and Principal Partition of a Linear Graph”. In: *IEEE transactions on circuit theory* 16.3 (1969), pp. 323–330. ISSN: 0018-9324.

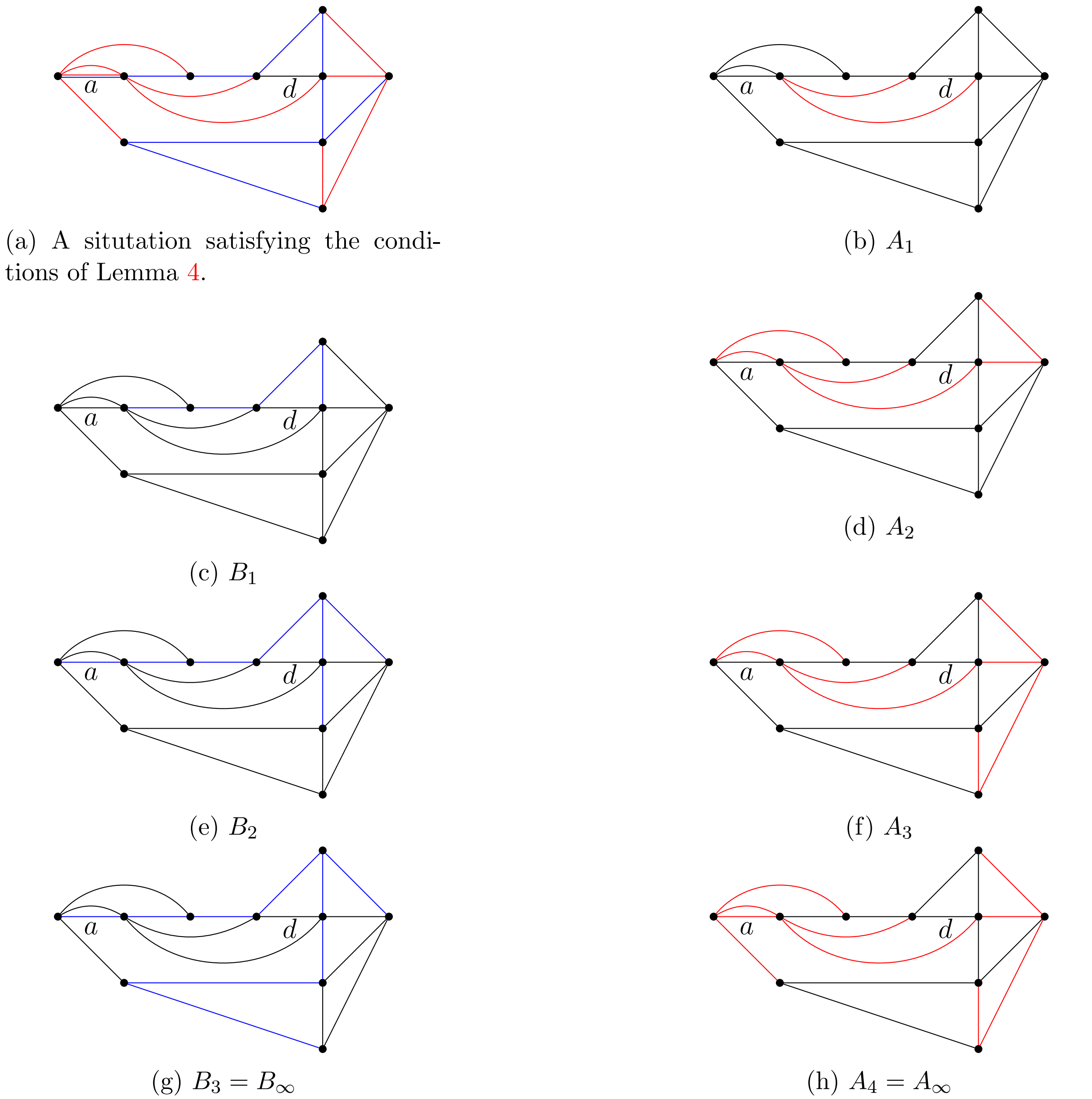
[11] Eugene Lawler. *Combinatorial Optimization: Networks and Matroids*. 1st. USA: Holt, Rinehart and Winston, 1976. ISBN: 0-03-084866-0.

[12] Alfred Lehman. “A Solution of the Shannon Switching Game”. In: *Journal of the Society for Industrial and Applied Mathematics* 12.4 (1964), pp. 687–725. ISSN: 0368-4245.

[13] Rune K. Rasmussen. “Algorithmic approaches for playing and solving Shannon games”. PhD thesis. Queensland University of Technology, 2008. URL: https://eprints.qut.edu.au/18616/.

A Example of the situation in Lemma 4 where k is not unique

Figure 17 shows an example of the situation in Lemma 4 where Lehman's $|a|$ from his Lemma 11 is not uniquely defined. This is due to his definition of A_{i+1}: the second 'part', i.e. $(B_i - \text{sp}(A_i)) \cap A$, indicates that every branch $e \in B_i$ that is in A should be included in A_{i+1}, *except* when it was already spanned by A_i. This condition was probably meant to not have to check all branches that were already in (the span of) A, which is often unnecessary. In special cases such as the situation on the next page, however, a is both spanned by A_2 and is a member of A. This causes a not to be included in A_3, and hence we have both $a \in B_2 - A_2$ and $a \in B_3 - A_3$. The example is constructed in such a way that we still have $a \in A_\infty$, which was assumed.



(a) A situtation satisfying the conditions of Lemma 4.

(b) A_1

(c) B_1

(d) A_2

(e) B_2

(f) A_3

(g) $B_3 = B_\infty$

(h) $A_4 = A_\infty$

Figure 17: An example where Lehman's $|a|$ is not uniquely defined (or alternatively: where our k is not unique).